Beyond Random Sampling: Efficient Language Model Pretraining via Curriculum Learning

Anonymous ACL submission

Abstract

Curriculum learning—organizing training data from easy to hard—has improved efficiency across machine learning domains, yet remains underexplored for language model pretraining. We present the first systematic investigation of curriculum learning in LLM pretraining, with over 200 models training across three strategies: vanilla curriculum learning, pacing-based sampling, and interleaved curricula, guided by six difficulty metrics spanning linguistic and information-theoretic properties. We evaluate performance on eight benchmarks under three realistic scenarios: limited data, unlimited data, and continual training. Our experiments show that curriculum learning consistently accelerates convergence in early and mid-training phases, reducing training steps by 18-45% to reach baseline performance. When applied as a warmup strategy before standard random sampling, curriculum learning yields sustained improvements up to 3.5%. We identify compression ratio, lexical diversity (MTLD), and readability (Flesch Reading Ease) as the most effective difficulty signals. Our findings demonstrate that data ordering—orthogonal to existing data selection methods—provides a practical mechanism for more efficient LLM pretraining.

1 Introduction

Large language models (LLMs) have achieved remarkable progress across diverse NLP tasks (Kaplan et al., 2020; Achiam et al., 2023; Anil et al., 2023). Yet, this progress comes at enormous computational and data costs, driving increasing interest in methods that make pretraining more efficient. Many efforts focus on improving data quality through filtering (Tirumala et al., 2023; Sorscher et al., 2022; Longpre et al., 2024) and optimizing data mixtures (Xie et al., 2023; Sachdeva et al., 2024). However, the ordering of training data remains largely overlooked. Unlike humans, who learn progressively from simple to complex concepts, LLMs are typically trained on randomly sampled data — a process that ignores the possible benefits of structured learning.

Among potential approaches, curriculum learning (CL) (Bengio et al., 2009a) offers a promising framework: by presenting training data in a structured progression from easy to hard examples, it mimics human learning process and has been shown to improve optimization and generalization in various domains like computer vision (Kumar et al., 2010; Sinha et al., 2020). This structured learning approach aims to achieve two important benefits: (i) faster convergence, and (ii) improved best model performance (Bengio et al., 2009a).

Despite its intuitive motivation, curriculum learning remains underexplored in the context of LLM pretraining, due to key challenges in defining difficulty measures and designing effective curriculum schedulers for large-scale training (Soviany et al., 2022). In this work, we seek to answer the question: *How can we build effective curriculum learning strategies for LLM pretraining?*

To address this question, we conduct the first systematic investigation of curriculum learning for LLM pretraining. We evaluate three curriculum strategies **(i) vanilla curriculum learning** of a fixed dataset (Bengio et al., 2009a), **(ii) sampling guided by pacing functions** (Wu et al., 2020; Nagatsuka et al., 2023), and **(iii) interleaved curricula** (Yang et al., 2024) that mix difficulty levels during training. We select 6 difficulty metrics out of 15 candidates through correlation analysis to characterize training data from multiple linguistic and information-theoretic perspectives. Using the CulturaX dataset (Nguyen et al., 2024), we train a set of 0.5B - 3B parameter LMs under three representative training scenarios: **limited data training**, **unlimited data training**, and **continual training**. We evaluate them on eight established benchmarks, spanning commonsense reasoning, language understanding, and reading comprehension.

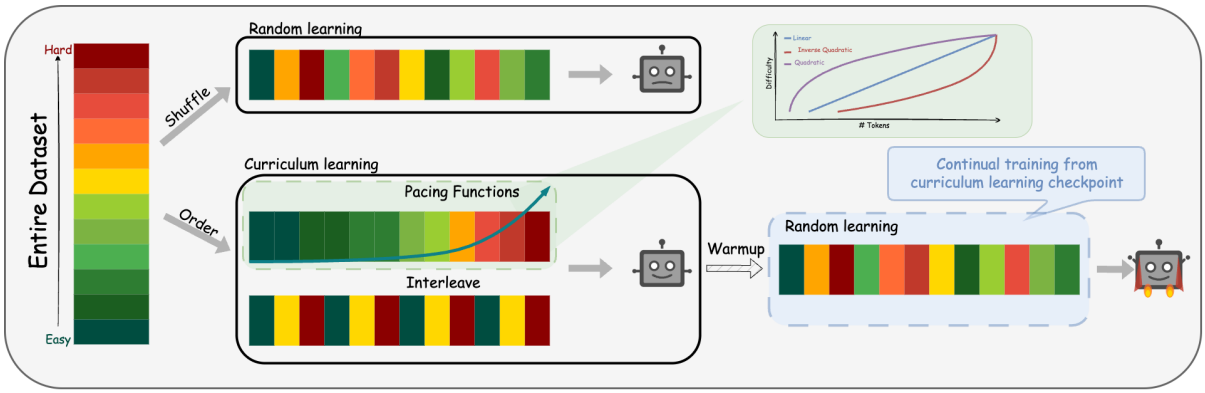

Figure 1: Curriculum learning for LLM pretraining organizes data from easy to hard based on a difficulty metric. This can involve strict ordering or progression governed by pacing functions such as linear, quadratic, or inverse quadratic. Alternatively, interleaved sampling strategies mix difficulty levels within each training segment. These strategies aim to enhance data efficiency and convergence. When using curriculum learning as warm-up followed by random training can further improve performance.

Our findings demonstrate that curriculum learning can provide consistent advantages over random baselines under certain combinations of difficulty metrics and curriculum strategies, particularly in the early and mid phases of training. Models trained with curricula reached baseline performance 18–45% faster, and we show that curriculum-based warmup can yield lasting performance gains even when followed by randomly sampled training with up to 3.5% improvement. These results underscore the potential of curriculum design to enhance pretraining efficiency and open new directions for scalable, data-aware model development.

Our contributions are threefold: (1) a comprehensive study of curriculum learning in LLM pretraining under three realistic training scenarios with five data ordering strategies; (2) an empirical analysis of six difficulty metrics and their effects on model convergence and performance; and (3) evidence that curriculum-based warmup can serve as a practical mechanism for efficient model training. Our work provides actionable insights for improving the efficiency of LLM pretraining in both academic and industrial settings.

2 Related Work

Data-Efficient LLM Training Recent work on data-efficient LLM pretraining emphasizes pruning, reweighting, and selection to reduce training costs without sacrificing performance. Perplexity-based filtering (Marion et al., 2023), robust domain mixing (DoReMi) (Xie et al., 2023), and embedding or influence-based sampling (Tirumala et al., 2023; Yu et al., 2024) all yield strong efficiency gains. Model-driven approaches like ASK-LLM and DENSITY leverage quality and diversity signals to outperform full-data baselines with fewer tokens (Sachdeva et al., 2024). Unlike these methods, which rely on sampling or reweighting data, we explore curriculum-based ordering over a pre-filtered dataset, making our approach complementary and orthogonal to prior works, providing a new dimension for improving training efficiency.

Curriculum Learning Curriculum learning (CL), introduced by (Bengio et al., 2009b), improves convergence by training on increasingly difficult data. Early NLP applications include grammar induction and machine translation (Spitkovsky et al., 2009; Zhang et al., 2018). Later work extended CL to LSTMs (Cirik et al., 2016) and transformers (Nagatsuka et al., 2023), though often limited to masked language modeling. Recent studies apply CL during LLM fine-tuning (Yang et al., 2024), or explore data-efficient schedules via skill learning and model preference (Chen et al., 2023; Zhang et al., 2025). Other strategies vary input lengths or attention to reduce compute (Pouransari et al., 2024; Kim and Lee, 2024). However, prior work largely focuses on fine-tuning or narrow CL setups. Our work is the first to systematically study CL during LLM pretraining, evaluating multiple paradigms and difficulty metrics at scale.

Text Difficulty Estimation Numerous studies explore text difficulty and quality in various applications. Length-based heuristics and term frequency are common proxies for complexity (Nagatsuka

et al., 2023; Spitkovsky et al., 2009; Liu et al., 2018). Lexical diversity metrics like MTLD and vocd-D provide finer-grained signals, with MTLD noted for its robustness across corpora (McCarthy and Jarvis, 2010). Information-theoretic measures such as compression ratio and entropy capture redundancy and quality (Yin et al., 2024), while recent work introduces perplexity-based preference modeling and diversity coefficients to assess conceptual variability (Zhang et al., 2025; Miranda et al., 2023). Together, these works provide valuable candidates for exploring difficulty metrics in curriculum learning.

3 Methodology

In this section, we discuss our choice of difficulty metrics, training scenarios, as well as the curriculum learning settings for our experiments.

3.1 Difficulty Metrics

To apply curriculum learning to LLM pretraining, we first constructed a diverse pool of 15 metrics, categorized into six conceptual dimensions: information density, lexical diversity, readability, fertility, model-perceived difficulty, and sequence length. To ensure orthogonality and avoid redundancy, we performed a Spearman correlation analysis (see Appendix A) across these candidates. Unlike previous works, we choose metrics that are easy to compute, independent of the choice of model and language, and have minimal computation overhead for model training (see Appendix H.

From this analysis, we selected six representative and minimally correlated metrics that capture distinct aspects of textual complexity: **Compression Ratio** (Yin et al., 2024), measuring how compactly information is encoded, serving as a proxy for redundancy and structural regularity; **Fertility** (Ali et al., 2024), Measures tokenization complexity, defined as the average number of subword tokens per word; **Flesch Reading Ease** (Kincaid, 1975), capturing readability by estimating text comprehensibility; Measure of Textual Lexical Diversity **(MTLD)** (McCarthy and Jarvis, 2010), captures lexical richness, offering a length-insensitive and robust estimate of vocabulary diversity; **Number of Tokens**, token level sequence length; **Perplexity**, a model-centric metric reflecting linguistic uncertainty as perceived by a pretrained language model.

3.2 Vanilla Curriculum Learning

We define our first training scenario of limited data training:

Scenario 1 (S1) *We assume to have access to a limited, fixed set of pretraining data, all of which must be utilized during training.*

Under S1, we adopt vanilla CL: a strict ordering of training samples from easy to hard (Bengio et al., 2009a). To do this, we sort the data by its difficulty score in ascending order. The model is then trained on this ordered set. This setting introduces more difficult samples at every step, aiming to gradually increase the model's capacity to handle complexity by building on previously acquired knowledge.

3.3 Curriculum Learning with Pacing

To introduce pacing functions, we define our second training scenario of training from a large data pool:

Scenario 2 (S2) *We assume to have access to a large (effectively unlimited) dataset, from which we are allowed to sample training data up to a fixed training budget.*

Under S2, instead of random sample training data, we sample data progressively using pacing functions that control the difficulty distribution over time.

Following works like (Hacohen and Weinshall, 2019; Nagatsuka et al., 2023), we split the dataset into N equally sized difficulty groups. At each training stage, the pacing function determines the number of samples drawn from each group. Sampling is performed randomly within groups to preserve diversity, while the pacing schedule governs the gradual transition from easy to hard examples.

We explore three types of pacing functions, as shown in Figure 1. Assume we have N difficulty groups, let T denote the total number of training tokens and t_i the number of tokens allocated to group i.

Linear Pacing.

$$t_i = \frac{T}{N} \tag{1}$$

The linear pacing function increases the difficulty level at a constant rate over time by an equal token allocation across groups.

Quadratic Pacing.

$$t_i = T \cdot \frac{(i+1)^2}{\sum_{j=1}^{N}(j+1)^2} \tag{2}$$

The quadratic pacing function allocates a larger portion of the token budget to higher-difficulty groups in later stages. Compared to linear pacing, this results in a faster increase in difficulty over training.

Inverse Quadratic Pacing.

$$t_i = T \cdot \frac{(N-i)^2}{\sum_{j=0}^{N-1}(N-j)^2} \tag{3}$$

Opposite to the quadratic approach, the inverse quadratic pacing function allocates more tokens to easier groups early on.

3.3.1 Interleaved Curriculum Learning

As a special case of pacing functions, we applied interleaved sampling. Rather than segmenting training into distinct difficulty phases, interleaved curricula maintain a continuous mix of difficulty levels throughout training as shown in Figure 1. The dataset is divided into N equally sized difficulty groups, and training is organized into I interleaves, each covering a token budget of T/I, where T is the total training budget. Within each interleave, we apply a linear pacing function to sample data in an easy-to-hard order, so for the current group i in interleave j, the number of tokens sampled $t_{i,j}$ is:

$$t_{i,j} = \frac{T}{I \cdot N} \tag{4}$$

This setup allows the model to repeatedly encounter the full difficulty spectrum while preserving a progressive structure within each interleave. Such exposure aims to improve generalization by preventing overfitting to a narrow band.

3.4 Experimental setting

Dataset We use the English subset of CulturaX (Nguyen et al., 2024) for pretraining, selected for its quality, transparency, and robust preprocessing. CulturaX is created by extensive cleaning: language ID, URL filtering, metric-based heuristics, document refinement, and fuzzy deduplication via MinHash. These steps are guided by large-scale data inspection and diverse quality metrics. We use the dataset as released, without further preprocessing.

Model We adopt a 0.5 billion parameter decoder-only language model for the main experiments based on the LLaMA3.2 (Grattafiori et al., 2024) architecture. The model comprises 16 transformer decoder layers with a hidden size of 1536 and 16 attention heads, using Multi-Query Attention (MQA) with 8 key-value groups. We apply rotary positional embeddings with a scaling factor of 1.0, $\theta = 10^5$, and a rotary dimension of 96. The feed-forward network uses a SwiGLU activation and an intermediate size of 4096, and a maximum sequence length of 2048. For the 1B and 3B models, we use the same configuration as LLaMA3.2.

Training We train all models using the AdamW optimizer (Loshchilov and Hutter, 2017) with $\beta_1 = 0.9$, $\beta_2 = 0.95$, weight decay of 0.1, and a fixed learning rate of 2×10^{-4}. We apply gradient clipping with a max norm of 1.0, mixed-precision training (cbfloat16), and dynamic loss scaling. The global batch size is 1,081,344 tokens. Training is performed on the Condor Galaxy 2 AI supercomputer (see Appendix F), and evaluations are run on NVIDIA A10 GPUs.

To ensure reproducibility across hardware, we conducted additional experiments comparing models trained on both platforms and observed consistent performance (Appendix F).

Evaluation We evaluate each model using LM-Eval (Gao et al., 2024), on a comprehensive set of 8 benchmarks from different categories, we report the average of all benchmarks. The details of the benchmarks are listed below:

- **Commonsense Reasoning**: PIQA (Bisk et al., 2020), COPA (Gordon et al., 2012), OpenBookQA (Mihaylov et al., 2018).
- **Language Understanding**: Hellaswag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2021), xwinograd_en (Tikhonov and Ryabinin, 2021).
- **Reading Comprehension**: BoolQ (Clark et al., 2019).
- **World Knowledge**: ArcChallenge (Clark et al., 2018).

All benchmarks are evaluated with 0-shot.

4 Experiments

In this section, we show our experiments in detail on the curriculum settings and report the average evaluation accuracy, since training or validation loss reflects token prediction accuracy on current dataset but poorly captures generalization, often showing weak correlation with downstream task performance (Liu et al., 2023; Hu et al., 2024). In contrast, benchmark evaluations directly assess

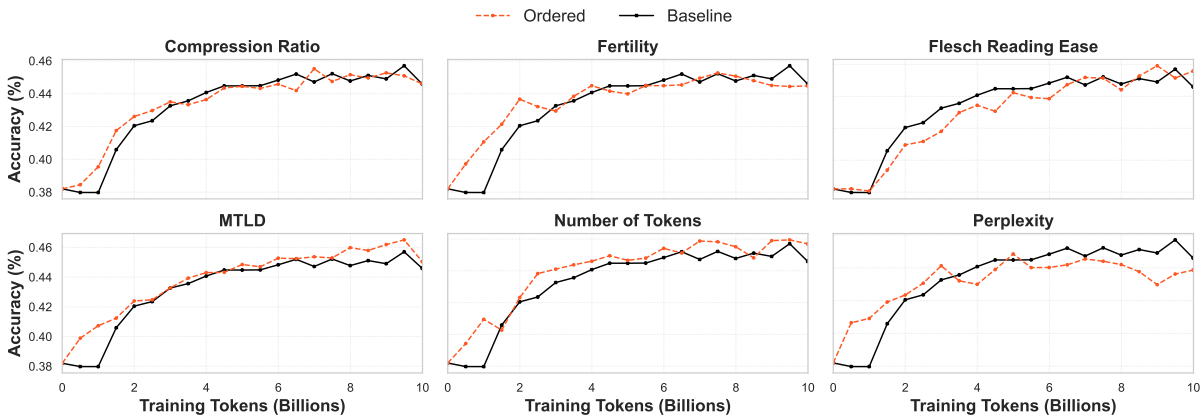

Figure 2: Evaluation accuracy of models trained with vanilla curriculum learning across 6 difficulty metrics, compared to a randomly shuffled baseline. CL settings offer a notable performance boost in early to mid stages, particularly under MTLD and Number of Tokens metrics, the models consistently outperform the baseline.

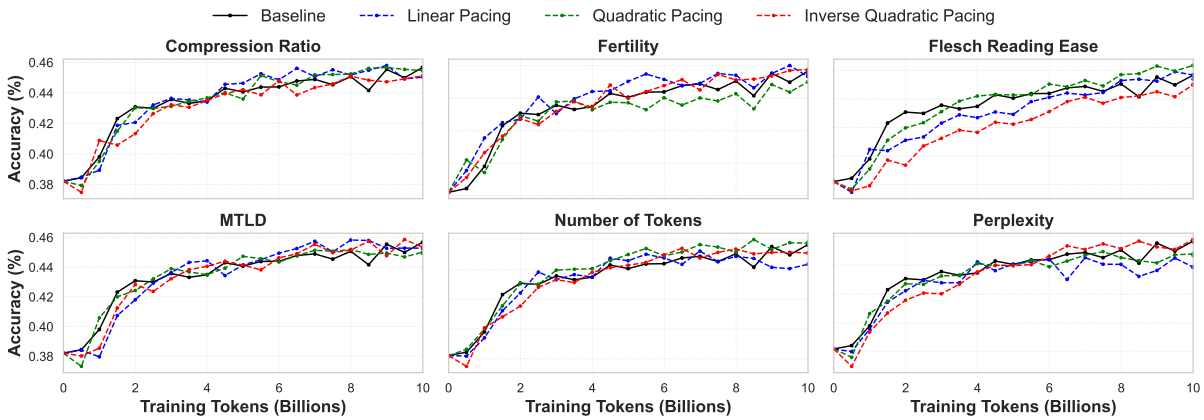

Figure 3: Evaluation accuracy of models trained with pacing-based curriculum learning across 3 pacing functions and 6 difficulty metrics. Compression Ratio, MTLD, and Fertility benefit most from linear pacing. Flesch Reading Ease and Number of Tokens yield best performance with quadratic pacing, with Flesch Reading Ease showing a consistent upward trend across all pacing functions.

real-world capabilities like reasoning and comprehension, offering a more reliable basis for comparing LLMs (Tay et al., 2021).

4.1 Experiment 1: Strict ordering

Setup Under S1, we first construct a fixed training set. Following the scaling laws of model performance (Kaplan et al., 2020), we sample 10B tokens from CulturaX to pretrain a 0.5B parameter model. For the 6 selected difficulty metrics, we construct one set for each by ordering the data based on the difficulty score. As the baseline, we randomly shuffled this fixed dataset. We pretrain one model for each of the 7 training sets.

Results *At the early stage of training, easier samples—according to most difficulty metrics—facilitate more efficient learning.*

Figure 2 presents the average accuracy of all benchmarks for models trained with strict ordering under 6 difficulty metrics, compared to the baseline. Except for Flesch Reading Ease, models trained on CL subsets consistently outperform the random baseline during the early stages of training (up to approximately 4B tokens). The advantage of curriculum learning narrows as training progresses, with performances converging or remaining slightly higher than the baseline by the end of training.

Among the 6 metrics, MTLD and Number of Tokens settings maintain benefits throughout the training, with MTLD achieving 1.8% higher best performance than baseline using 17.9% fewer training steps, and Number of Tokens achieves similar performance but with 27.5% fewer steps.

Interestingly, the Flesch Reading Ease setting

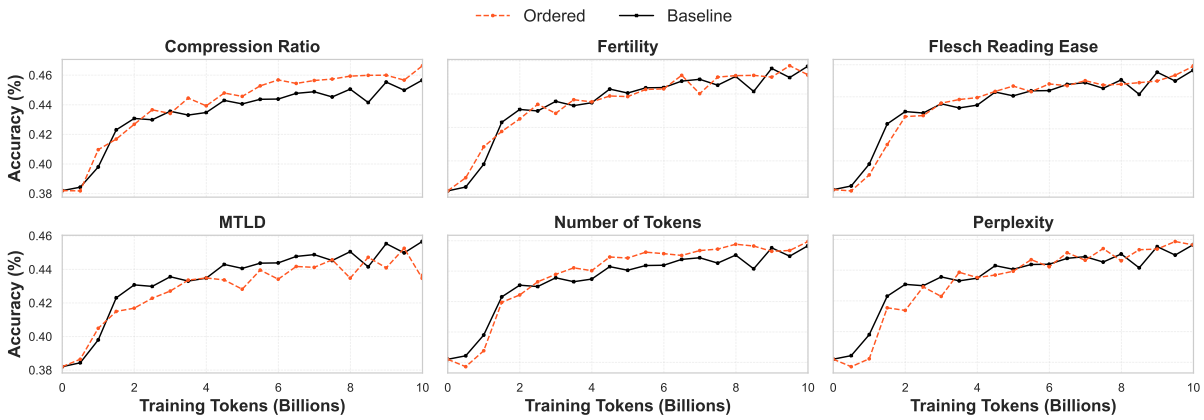

Figure 4: Evaluation accuracy of models trained with interleaved curriculum learning across 6 difficulty metrics. Interleaving consistently boosts performance for Compression Ratio and Number of Tokens, with the rest similar or slightly worse than the baseline.

initially underperforms the baseline but gradually catches up at a constant rate in later stages, showing a similar pattern as we observed in the following experiments, where we will provide more analysis.

Perplexity-based ordering, although showing strong early gains, exhibits a noticeable drop in performance during the later phases. We hypothesize that high-perplexity samples that are often both difficult and noisy, which could degrade the model's learning stability.

4.2 Experiment 2: Pacing Functions

Setup In this setting, we first follow S2 to randomly sample and shuffle 10B tokens as the baseline. Then for each difficulty metric, we first partition the dataset into 10 difficulty groups of equal size. We then apply linear, quadratic, and inverse quadratic pacing functions to sample data from each group, gradually building the training set to 10B tokens. We train one model for each combination of pacing function and difficulty metric.

Results *Compared with vanilla curriculum learning, a steady, progressive exposure to increasingly difficult groups while keeping variation within groups improves model performance more effectively for metrics capturing linguistic richness. And model trained on data ordered by Flesch Reading Ease show consistent improvement without a sign of convergence.*

Figure 3 shows results grouped by difficulty metrics. CL settings differs more to the baseline in early to mid training stages, which aligns with the findings in section 4.1.

Specifically, Compression Ratio, Fertility, and MTLD—metrics tied to linguistic richness and redundancy—benefit most from linear pacing, especially in the mid-training stage. Compression Ratio reaches baseline peak accuracy with 39.5% fewer steps; MTLD with 31.7% fewer steps; and Fertility achieves 99.5% of best baseline performance using 45.3% fewer steps.

For the Number of Tokens setting, all pacing strategies outperform the baseline between 3B–8B tokens, with quadratic pacing delivering the most stable gains: it achieves the baseline's performance in 29.8% fewer steps, reaches 99.5% of the peak in 44.8% fewer steps, and ultimately surpasses the baseline with a final score that is 1.1% higher.

For Flesch Reading Ease, interestingly, 3 pacing functions show a similar trend as shown in Experiment 1: model performance increases with constant speed and doesn't show a sign of convergence, with quadratic pacing yielding the best result: 1.6% above the baseline at the final checkpoint.

In contrast, Perplexity performs best under inverse quadratic pacing, while linear and quadratic pacing fall behind, likely due to noisy high-perplexity samples—consistent with findings from Experiment 1.

4.3 Experiment 3: Interleave Curriculum Learning

Setup In this setting, we apply interleaved curriculum learning under S2. As before, for each difficulty metric, the dataset is partitioned into 10 equal-sized difficulty groups. We then construct 10 interleaved subsets, each consisting of 1B tokens sampled linearly across all difficulty groups. These subsets are concatenated to form a 10B-token train-

ing set, allowing the model to encounter a mix of difficulties throughout training while preserving a progressive structure within each interleave.

Results *Only the Compression Ratio and Number of Tokens settings showed a consistent advantage over the baseline, while others remain similar or have slightly worse performance*

Figure 4 reports the averaged accuracy. Our results show that the compression ratio setting benefits the most from interleaved CL, where the model outperforms the baseline throughout the training by a large margin, with 2.2% higher performance achieved, and the best random baseline performance is reached with 41.7% fewer steps. A similar pattern can be seen for the Number of Tokens setting, where the model consistently outperforms the baseline and reached the best baseline performance with 23.5% less data.

In contrast, other metrics yield nearly the same performance as the random baseline and the MTLD setting has worse performance during training.

4.4 Curriculum Learning as model warmup

Inspired by the fact that the curriculum learning generally speeds up the early to mid training stage and training with CL is usually one-shot: once the training set is organized by CL, it is difficult to add new data to the dataset unless we reorganize the enlarged dataset again. So we consider the third training scenario of continual training:

Scenario 3 (S3) *The training consists of multi phases, where in the first phase the model is trained with a data budget, for later phases it can be further trained on more data.*

The S3 involves cases like continual training where CL can only apply to the first phase because organizing data from easy to hard is one-shot, and massive data training where organizing the entire dataset with CL is computationally hard.

Setup In these cases, we consider CL as a warmup phase for model training where we first train the model with a fixed data organized under CL, then continue training it with more data under the conventional training with shuffling.

For the CL training phase, we choose 3 CL settings that previously showed promising results: Compression Ratio, Number of Tokens with vanilla CL and Flesch Reading Ease with quadratic pacing. For the second training phase, we continue training the models of each setting as well as the

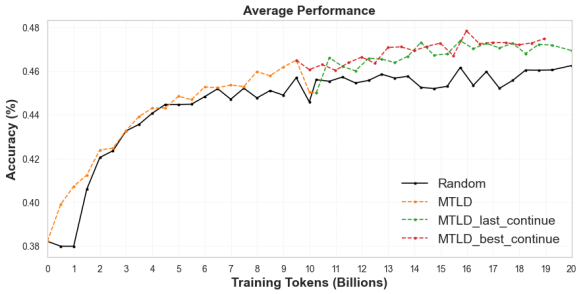

Figure 5: Performance of models trained with curriculum learning as a warmup phase, followed by continued training on randomly shuffled data, using MTLD as the difficulty metric, both best and last checkpoint warmups yield sustained advantages over the baseline.

baseline using the same 10B new data which is randomly shuffled. For curriculum learning settings, we choose two checkpoints to perform continual training: the best one and the last one, if they are not the same.

Results *Using CL as warmup consistently improves performance in later training phases by a large margin, allowing the model reach a higher convergence point.*

We show the result for the MTLD setting in Figure 5: models with a transition from CL to random training maintain a consistent performance advantage over the baseline. While the baseline performance converges, the warmup setting keeps improving regardless of starting from the best or last checkpoint. Specifically, the starting from the best and last lead to 3.5% and 2.6% higher than the baseline, respectively, and the baseline never surpasses the first phase of CL training with double the data. More results are in Appendix B.

5 Ablation Study

5.1 Number of difficulty groups

In our main experiments, we chose 10 difficulty groups when splitting the dataset to balance smooth difficulty transitions with sufficient intra-group diversity for effective learning.

Following (Nagatsuka et al., 2023), we split the data into 3 difficulty groups as easy, middle, and hard, to study the impact of fewer groups on the linear pacing case. Detailed results are shown in Appendix C, where using fewer groups leads to nearly equal or worse model performance.

We also tested the case with 20 groups in the early stage of training, showing no improvement compared to 10 groups. This indicates that when

applying pacing functions, the results are less sensitive to the choice of the number of groups. We assume fewer groups lead to coarser transitions and potential overfitting, while more groups exhibit diminishing returns, behaving similarly to strict ordering. Thus, we use 10 groups in our experiments, which act as a balance between smooth difficulty transitions and sufficient intra-group diversity.

5.2 Robustness of Curriculum Learning

To evaluate the robustness of our curriculum learning strategies, we repeat the baseline and key experiments across six high-performing settings—spanning different difficulty metrics and pacing types—twice, independently for all stages from data sampling to model training. A detailed analysis is provided in the Appendix D.

We observe consistent trends across runs, *the previous findings with the CL settings still hold on average, even when accounting for variability across runs.* This low variance across repetitions confirms that the observed gains are stable and not due to chance, demonstrating that curriculum learning is a reliable approach for improving LLM pretraining efficiency.

5.3 Data sampling vs Curriculum Learning

We observed that in some settings, the model converges faster than the baseline but shows a performance drop at the last steps. The best practice is to select the best checkpoint. However, in CL, later-stage data can sometimes act as noisy or overly difficult samples, which may not further improve the model. So we ask the following question: *are the early convergence or performance peaks results from the data filtering effect of CL?*

To answer the question, we conduct additional experiments on selected settings that exhibit early convergence: MTLD, Number of Tokens with vanilla CL and Compression Ratio with linear pacing. Specifically, we extract training data used before the best-performing checkpoint, randomly shuffle it, and train the model again on this shuffled subset. A detailed comparison of the model's performance on the shuffled subset versus the original ordered setting is provided in the Appendix E.

Our results show that using *the early stage data of curriculum learning indeed act as a data filtering and lead to improvement compared with the random baseline, but still fall behind the ordered data setting*, showing the effectiveness of data ordering during the training of the model.

5.4 Scale Curriculum Learning

In order to investigate how well our findings scale up with model size, we train a set of the 1B and 3B models with 20B and 40B tokens respectively on: (i) the vanilla CL setting, (ii) Flesch Reading Ease with quadratic pacing and (iii) CL as warmup from MTLD vanilla CL and Flesch Reading Ease with quadratic pacing setting. Full training configurations and results are presented in Appendix G.

Our results indicate that *the benefits of curriculum learning extend to larger models.* In particular, the warmup setting consistently delivers sustained performance improvements throughout training, demonstrating the generalizability of curriculum learning across model scales and training scenarios.

6 Discussion and Conclusion

Our work presents the first systematic exploration of curriculum learning (CL) for large language model pretraining, comparing vanilla, pacing-based, and interleaved curricula across six difficulty metrics. We demonstrate that **CL consistently accelerates convergence—reducing training steps by 18–45%—and improves final performance by up to 3.5% when applied as a warmup before random sampling.** This shows that CL not only speeds early and mid-stage training but also yields lasting gains in later phases.

Among the examined metrics, **Compression Ratio, MTLD, and Flesch Reading Ease stand out as the most effective indicators, linking linguistic diversity and information density to better learning progress.** Linear pacing and interleaved strategies further enhance generalization by gradually and repeatedly exposing the model to increasing difficulty levels.

Importantly, **our approach introduces no computational overhead**—difficulty metrics are lightweight, model- and language-independent, and can be precomputed efficiently. CL operates purely on data ordering, orthogonal to data selection or pruning techniques, and thus integrates seamlessly into existing pretraining pipelines.

Overall, curriculum learning offers a simple yet powerful mechanism to improve training efficiency at scale. It can achieve both faster training and improved performance, even be used as a warmup phase, paving the way toward more data- and compute-efficient LLM development. Future work should explore adaptive and model-aware curricula for even greater efficiency gains.

Limitations

This work focuses on LLM pretraining via curriculum learning by training 0.5B and 1B parameter models on various combinations of curriculum strategies and difficulty metrics. While our findings consistently demonstrate the benefits of curriculum learning (CL) across training scenarios, several limitations remain:

First, our experiments are constrained to decoder-only architectures (LLaMA3.2-like models) and English-language data (CulturaX subset), which may limit the generalizability of our results to encoder-based models or multilingual settings. Future work should evaluate CL across diverse model families and linguistic contexts.

Second, we apply static, precomputed difficulty scores and do not explore adaptive or dynamic curricula that respond to the model's evolving capabilities. More sophisticated scheduling techniques, such as model-aware or task-informed pacing, could yield further gains.

Third, while we test six difficulty metrics, our metric selection is based on correlation analysis, and some potentially useful signals (e.g., syntactic depth or coherence) were not explored. Additionally, metrics like perplexity may conflate difficulty with noise, reducing their reliability.

Fourth, this work focuses exclusively on the pretraining phase and does not examine the impact of curriculum learning on downstream fine-tuning. Evaluating how curriculum-pretrained models transfer to various fine-tuning regimes remains an important direction for future study.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Mehdi Ali, Michael Fromm, Klaudia Thellmann, Richard Rutmann, Max Lübbering, Johannes Leveling, Katrin Klug, Jan Ebert, Niclas Doll, Jasper Buschhoff, et al. 2024. Tokenizer choice for llm training: Negligible or crucial? In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3907–3924.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009a. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48, New York, NY, USA. Association for Computing Machinery.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009b. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.

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

Mayee Chen, Nicholas Roberts, Kush Bhatia, Jue Wang, Ce Zhang, Frederic Sala, and Christopher Ré. 2023. Skill-it! a data-driven skills framework for understanding and training language models. *Advances in Neural Information Processing Systems*, 36:36000–36040.

Volkan Cirik, Eduard Hovy, and Louis-Philippe Morency. 2016. Visualizing and understanding curriculum learning for long short-term memory networks. *arXiv preprint arXiv:1611.06204*.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Meri Coleman and Ta Lin Liau. 1975. A computer readability formula designed for machine scoring. *Journal of Applied Psychology*, 60(2):283.

Edgar Dale and Jeanne S Chall. 1948. A formula for predicting readability: Instructions. *Educational research bulletin*, pages 37–54.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2024. The language model evaluation harness.

Andrew Gordon, Zornitsa Kozareva, and Melissa Roemmele. 2012. SemEval-2012 task 7: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of*

the Sixth International Workshop on Semantic Evaluation (SemEval 2012), pages 394–398, Montréal, Canada. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Robert Gunning. 1969. The fog index after twenty years. *Journal of Business Communication*, 6(2):3–13.

Guy Hacohen and Daphna Weinshall. 2019. On the power of curriculum learning in training deep networks. In *International conference on machine learning*, pages 2535–2544. PMLR.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

Yutong Hu, Quzhe Huang, Mingxu Tao, Chen Zhang, and Yansong Feng. 2024. Can perplexity reflect large language model's ability in long text understanding? *arXiv preprint arXiv:2405.06105*.

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

Jisu Kim and Juhwan Lee. 2024. Strategic data ordering: Enhancing large language model performance through curriculum learning. *arXiv preprint arXiv:2405.07490*.

J.P. Kincaid. 1975. *Derivation of New Readability Formulas: (automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel*. Research Branch report. Chief of Naval Technical Training, Naval Air Station Memphis.

M Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-paced learning for latent variable models. *Advances in neural information processing systems*, 23.

Cao Liu, Shizhu He, Kang Liu, Jun Zhao, et al. 2018. Curriculum learning for natural answer generation. In *IJCAI*, pages 4223–4229.

Hong Liu, Sang Michael Xie, Zhiyuan Li, and Tengyu Ma. 2023. Same pre-training loss, better downstream: Implicit bias matters for language models. In *International Conference on Machine Learning*, pages 22188–22214. PMLR.

Shayne Longpre, Gregory Yauney, Emily Reif, Katherine Lee, Adam Roberts, Barret Zoph, Denny Zhou, Jason Wei, Kevin Robinson, David Mimno, et al. 2024. A pretrainer's guide to training data: Measuring the effects of data age, domain coverage, quality, & toxicity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3245–3276.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Max Marion, Ahmet Üstün, Luiza Pozzobon, Alex Wang, Marzieh Fadaee, and Sara Hooker. 2023. When less is more: Investigating data pruning for pretraining llms at scale. *arXiv preprint arXiv:2309.04564*.

G Harry Mc Laughlin. 1969. Smog grading-a new readability formula. *Journal of reading*, 12(8):639–646.

Bryan C McCannon. 2019. Readability and research impact. *Economics Letters*, 180:76–79.

Philip M McCarthy and Scott Jarvis. 2010. Mtld, vocd-d, and hd-d: A validation study of sophisticated approaches to lexical diversity assessment. *Behavior research methods*, 42(2):381–392.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*.

Brando Miranda, Alycia Lee, Sudharsan Sundar, Allison Casasola, and Sanmi Koyejo. 2023. Beyond scale: The diversity coefficient as a data quality metric for variability in natural language data. *arXiv preprint arXiv:2306.13840*.

Koichi Nagatsuka, Clifford Broni-Bediako, and Masayasu Atsumi. 2023. Length-based curriculum learning for efficient pre-training of language models. *New Generation Computing*, 41(1):109–134.

Thuat Nguyen, Chien Van Nguyen, Viet Dac Lai, Hieu Man, Nghia Trung Ngo, Franck Dernoncourt, Ryan A. Rossi, and Thien Huu Nguyen. 2024. CulturaX: A cleaned, enormous, and multilingual dataset for large language models in 167 languages. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4226–4237, Torino, Italia. ELRA and ICCL.

Hadi Pouransari, Chun-Liang Li, Jen-Hao Chang, Pavan Kumar Anasosalu Vasu, Cem Koc, Vaishaal Shankar, and Oncel Tuzel. 2024. Dataset decomposition: Faster llm training with variable sequence length curriculum. *Advances in Neural Information Processing Systems*, 37:36121–36147.

Brian Richards. 1987. Type/token ratios: What do they really tell us? *Journal of child language*, 14(2):201–209.

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

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Samarth Sinha, Animesh Garg, and Hugo Larochelle. 2020. Curriculum by smoothing. *Advances in Neural Information Processing Systems*, 33:21653–21664.

Edgar A Smith and RJ Senter. 1967. *Automated readability index*, volume 66. Aerospace Medical Research Laboratories, Aerospace Medical Division, Air . . . .

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

Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. 2022. Curriculum learning: A survey. *International Journal of Computer Vision*, 130(6):1526–1565.

Valentin I Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. 2009. Baby steps: How "less is more" in unsupervised dependency parsing. *NIPS: Grammar Induction, Representation of Language and Language Learning*, pages 1–10.

Yi Tay, Mostafa Dehghani, Jinfeng Rao, William Fedus, Samira Abnar, Hyung Won Chung, Sharan Narang, Dani Yogatama, Ashish Vaswani, and Donald Metzler. 2021. Scale efficiently: Insights from pre-training and fine-tuning transformers. *arXiv preprint arXiv:2109.10686*.

Alexey Tikhonov and Max Ryabinin. 2021. It's all in the heads: Using attention heads as a baseline for cross-lingual transfer in commonsense reasoning. *Preprint*, arXiv:2106.12066.

Kushal Tirumala, Daniel Simig, Armen Aghajanyan, and Ari Morcos. 2023. D4: Improving llm pretraining via document de-duplication and diversification. *Advances in Neural Information Processing Systems*, 36:53983–53995.

Xiaoxia Wu, Ethan Dyer, and Behnam Neyshabur. 2020. When do curricula work? *arXiv preprint arXiv:2012.03107*.

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. 2023. Doremi: Optimizing data mixtures speeds up language model pretraining. *Advances in Neural Information Processing Systems*, 36:69798–69818.

Yushi Yang, Andrew M Bean, Robert McCraith, and Adam Mahdi. 2024. Evaluating fine-tuning efficiency of human-inspired learning strategies in medical question answering. *arXiv preprint arXiv:2408.07888*.

Mingjia Yin, Chuhan Wu, Yufei Wang, Hao Wang, Wei Guo, Yasheng Wang, Yong Liu, Ruiming Tang, Defu Lian, and Enhong Chen. 2024. Entropy law: The story behind data compression and llm performance. *arXiv preprint arXiv:2407.06645*.

Zichun Yu, Spandan Das, and Chenyan Xiong. 2024. Mates: Model-aware data selection for efficient pretraining with data influence models. *Advances in Neural Information Processing Systems*, 37:108735–108759.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

Xuan Zhang, Gaurav Kumar, Huda Khayrallah, Kenton Murray, Jeremy Gwinnup, Marianna J Martindale, Paul McNamee, Kevin Duh, and Marine Carpuat. 2018. An empirical exploration of curriculum learning for neural machine translation. *arXiv preprint arXiv:1811.00739*.

Xuemiao Zhang, Liangyu Xu, Feiyu Duan, Yongwei Zhou, Sirui Wang, Rongxiang Weng, Jingang Wang, and Xunliang Cai. 2025. Preference curriculum: Llms should always be pretrained on their preferred data. *arXiv preprint arXiv:2501.13126*.

A Metrics Selection

To choose a set of difficulty metrics for our curricula, we began with fifteen candidate metrics spanning six conceptual categories: *Information Density* (Compression Ratio), *Lexical Diversity* (TTR (Richards, 1987), MTLD, HD-D (McCarthy and Jarvis, 2010)), *Readability* (Flesch Reading Ease, Flesch–Kincaid Grade (Kincaid, 1975), Coleman–Liau Index (Coleman and Liau, 1975), automated readability index(ARI) (Smith and Senter, 1967), Dale–Chall (Dale and Chall, 1948), Linsear Write (McCannon, 2019), Gunning Fog (Gunning, 1969), SMOG (Mc Laughlin, 1969)), *Sequence Length* (Number of Tokens), *Fertility* (Fertility Score), and *Perplexity* (Perlexity calculated by KenLM (Heafield, 2011))

We computed Spearman rank correlations on our pretraining data to assess monotonic relationships among these metrics. Results are presented in Figure 6. Strong positive correlations ($\rho \approx 1$) appear among the various readability formulas, indicating redundancy; moderate correlations ($\rho \approx 0.3$–0.5) cluster the diversity measures; and near-zero coefficients suggest orthogonality (notably between Perplexity or Fertility and most others).

Guided by these patterns, we retained one representative from each highly inter-correlated block and preserved metrics that capture unique aspects of text complexity. Our final six signals are Compression Ratio, Flesch Reading Ease, MTLD, Perplexity, Fertility, and Number of Tokens—ensuring broad coverage of linguistic and information-theoretic difficulty.

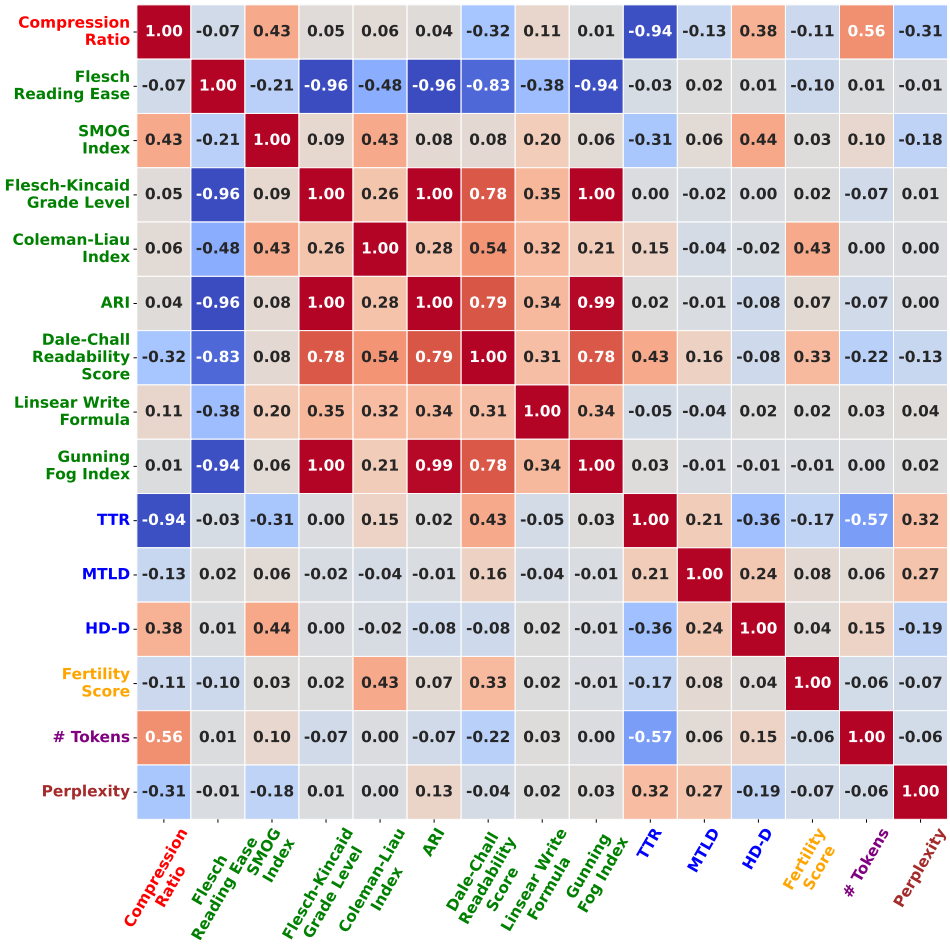

Figure 6: Spearman correlation matrix of the fifteen candidate metrics. Cell shading encodes correlation strength (red = strong positive; blue = strong negative), highlighting the correlations across the different metrics: Density (red), Readability (green), Diversity (blue), Lexical (purple), Fertility (orange), and Perplexity (brown).

B Curriculum Learning as Warmup

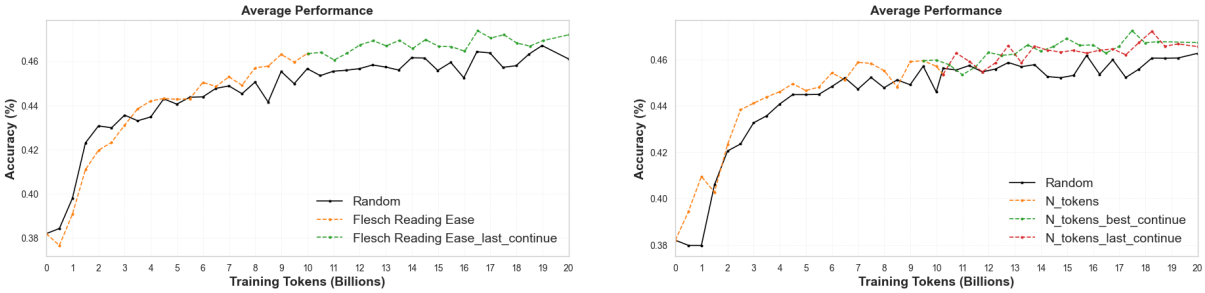

Figure 7: The average accuracy of models continually trained on two curriculum learning settings: Flesch Reading Ease with quadratic pacing (left) and Number of Tokens with vanilla CL (right), we select the best checkpoint and the last checkpoints (they can be the same) from the CL settings to start continual training.

In Figure 7 we show two more experiments that use curriculum learning as warmup: the Flesch Reading Ease with quadratic pacing and Number of Tokens with vanilla curriculum learning. The models trained on both settings and their baselines are then continually trained on randomly shuffled data.

For the model trained on Flesch Reading Ease with quadratic pacing setting, we continue training the model from the last checkpoint since it is also the best. We observed a similar pattern as in Figure 5: CL as warmup setting consistently outperforms the baseline by a large margin, especially in the continual training phase. Specifically, this setting achieves 1.5% higher accuracy and reaches the best baseline performance with 40.0% fewer steps.

For the model trained on Number of Tokens with vanilla CL setting, we chose both the best and last checkpoint to start continual training. The models trained from both checkpoints show close performance throughout training, with the model starting from the best checkpoint slightly better than the model starting from the last checkpoint. Overall the CL as warmup settings achieve 2.2% higher accuracy than the baseline and reached the best baseline performance with 20.7% fewer steps.

C Number of difficulty groups

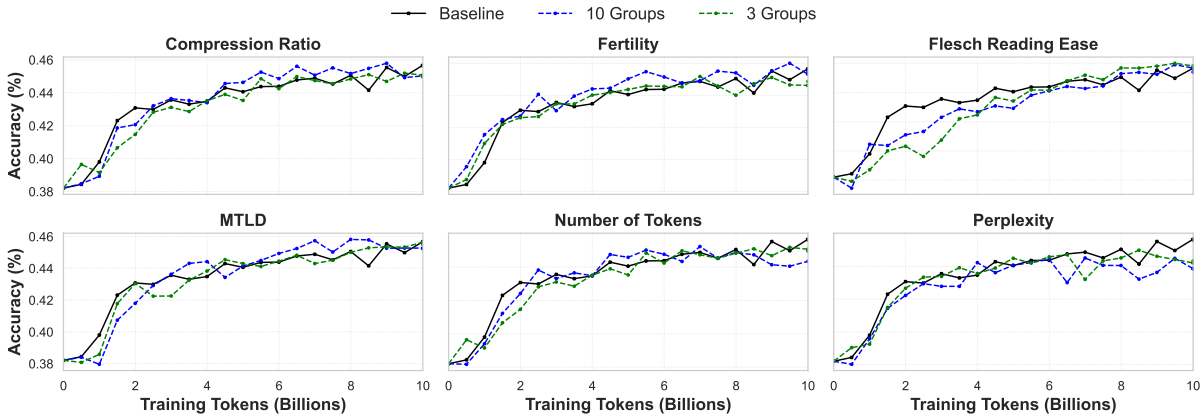

Figure 8: The comparison of the average accuracy for models trained via CL on 6 difficulty metrics under the linear pacing function with 3 and 10 difficulty groups.

We investigate the influence of using a different number of difficulty groups, and show the results when using 3 difficulty groups compared with our default 10 difficulty groups with the linear pacing CL setting in Figure 8. Though 3 difficulty groups can be seen as easy, middle and hard groups intuitively, the performance is generally worse or similar to 10 groups with a small difference. We hypothesize that 3 groups lead to coarser transitions between groups and potentially lead to model overfitting to each group,

which can be harmful for model training. But overall, we consider the result is less sensitive to the choice of the number of groups.

D Robustness of Curriculum Learning

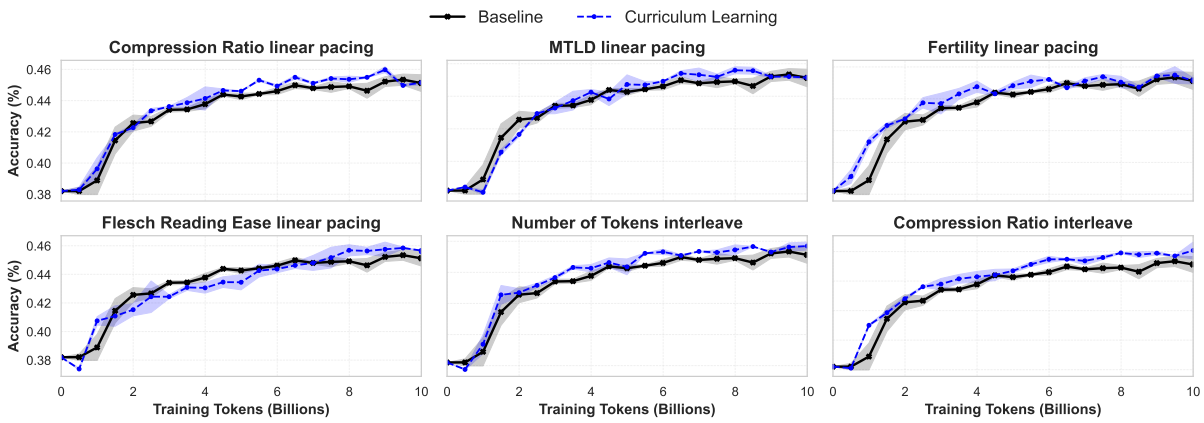

Figure 9: Performance comparisons between baseline (black) and curriculum learning (blue) across six high-performing settings. Accuracy is averaged over three runs, with shaded areas denoting standard deviations. Results confirm the robustness of curriculum learning, with consistent improvements and minimal overlap in variation with the baseline.

We select 6 settings that previously showed strong performance: Compression Ratio, MTLD, Fertility and Flesch Reading Ease with linear pacing; Number of Tokens and Compression Ratio with interleave pacing, and we repeat two more times for each setting, as well as the baseline. Figure 9 shows the results with average performance across runs and the standard deviations. For all tested settings, the previously established findings still hold on average, even considering the standard deviation, as there are very small part of overlapping between the variation range for CL settings and the baseline. This shows the robustness of our defined CL settings.

E Sampling vs Ordering

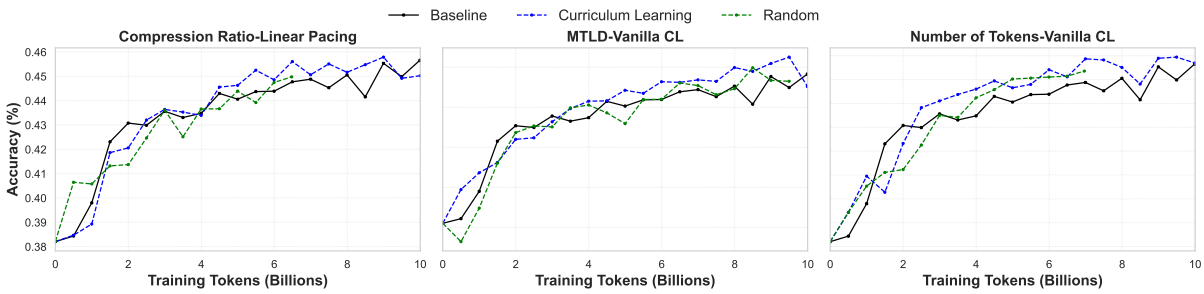

Figure 10: Performance comparisons between curriculum learning ordering and random shuffling of the data selected up to a selected difficulty level. Though leaving out the most difficult samples can be viewed as an effective data cleaning process, the performance is still lower than the ordered setting using the same data.

We conduct a controlled comparison to disentangle the effects of data selection from data ordering. We select 3 CL settings where the model reached best performance early but experience a performance drop at last steps: MTLD, Number of Tokens with vanilla curriculum learning and Compression Ratio with linear pacing. For each, we extract the training data used before the best-performing checkpoint, then randomly shuffle this subset and retrain the model. While these shuffled subsets outperform the full random baseline—suggesting a filtering effect—the ordered curriculum versions still achieve superior performance.

This indicates that the benefits of curriculum learning arise not only from selecting informative data but also from the order in which it is presented during training.

F Cerebras

CS-2 systems are purpose-built network-attached AI accelerators. Each CS-2 features 40 GB of SRAM and a peak of 62.5 AI PetaFLOPs, providing a total of 4 ExaFLOPs of AI compute across 64 systems in the CG-2 supercomputer. Utilizing the weight streaming mode of the Cerebras software stack, the Condor Galaxy supercomputers can flexibly schedule multiple jobs based on hardware resource requirements and priority. The number of CS-2s allocated to a job can be dynamically adjusted during training, with performance scaling linearly up to 64 CS-2s per job. This scalability is facilitated by the Cerebras software stack's use of pure data parallelism to distribute the workload across multiple CS-2s. Jobs are managed by a priority queue system, ensuring efficient allocation of computational resources.

MemoryX is a large-capacity off-wafer memory service used to store all model weights, gradients, and optimizer states. SwarmX is a broadcast/reduce fabric that connects the memory service MemoryX to each of the CS-2 systems in a wafer-scale cluster. SwarmX coordinates the broadcast of the model layer weights, giving each CS-2 a local copy, and it receives and aggregates (by addition) the independent weight gradients coming from the CS-2 systems during backpropagation. At the end of each iteration, the aggregated gradients are sent to MemoryX for weight update.

F.1 Reproducibility on Nvidia GPUs

To further eliminate the difference between models trained on the Cerebras System and Nvidia GPUs, we trained two 0.5B models with 10B tokens using Cerebras CS-2 System and Nvidia A100 GPU respectively, and evaluate the model performance on eight benchmarks, as shown in Table 1, we report the evaluation result for both models and the relative difference compared with model trained on Cerebras. Across all benchmarks, the relative difference is less than 0.5%, so we consider the model trained on both hardware makes no difference regarding model performance.

Benchmark	Cerebras	NVIDIA A100	$\Delta\%$ vs Cerebras
ArcChallenge	0.1928	0.1932	0.21
BoolQ	0.6101	0.6125	0.39
COPA	0.6900	0.6902	0.03
Hellaswag	0.3087	0.3092	0.16
OpenBookQA	0.1820	0.1823	0.16
PIQA	0.6561	0.6572	0.17
WinoGrande	0.4893	0.4903	0.20
xwinograd_en	0.5939	0.5948	0.15

Table 1: Performance comparison of NVIDIA A100 relative to Cerebras across multiple benchmarks.

G Scale Curriculum Learning

We evaluate the scalability of our findings by replicating key curriculum learning settings using 1B and 3B models following the Llama3.2 configuration. The training hyperparameters are the same as previous experiments.

G.1 Vanilla Curriculum Learning

In this section, we experiment with the vanilla CL setting, we use the same set of difficulty metrics and instead of selecting a dataset of 10B, we select 20B tokens according to the scaling law, and create 6 strictly ordered training set as well as a baseline set, we train a model for each set.

Figure 11 shows the results. We observe similar findings to our previous experiments: (1) the CL settings and the baseline differentiate the most in the early to mid training stages, with most CL settings reaching the best performance faster than the baseline; (2) the MTLD setting consistently outperforms the baseline during the training; (3) Number of Tokens setting has a faster performance improvement in the

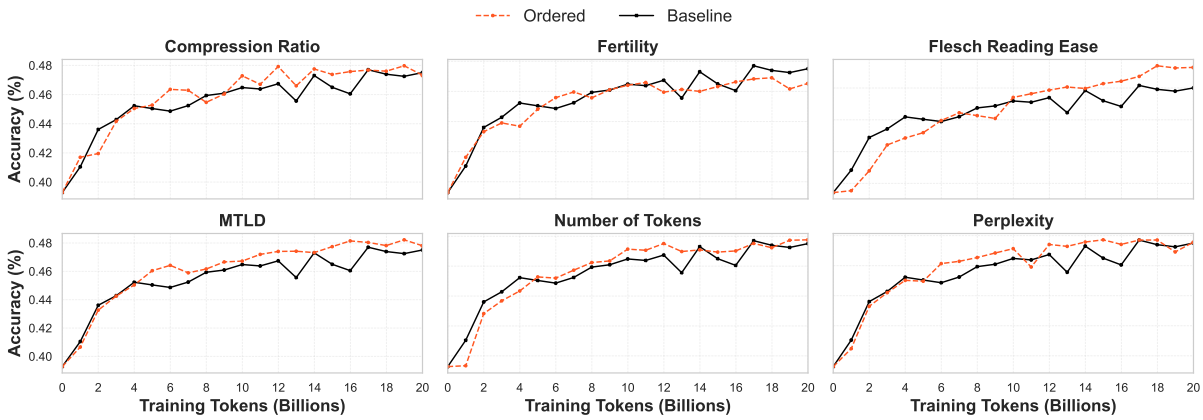

Figure 11: Performance of vanilla curriculum learning at scale using 1B model trained on 20B tokens compared to the baseline. Most settings show a faster improvement in the early to mid stage, with MTLD and Flesch Reading Ease showing a strong performance.

early and mid stage of training; (4) Flesch Reading Ease setting keeps a constant improvement rate of performance, interestingly, outperforms the baseline by a large margin at the end of training.

G.2 Flesch Reading Ease with quadratic pacing

Since the Flesch Reading Ease setting showed promising and unique results from all previous settings: a constant rate of model performance improvement and didn't show a sign of convergence. We want to test this specifically for a larger model, thus, we chose the setting that shows the biggest improvement rate and final performance-quadratic pacing-to train a 1B model.

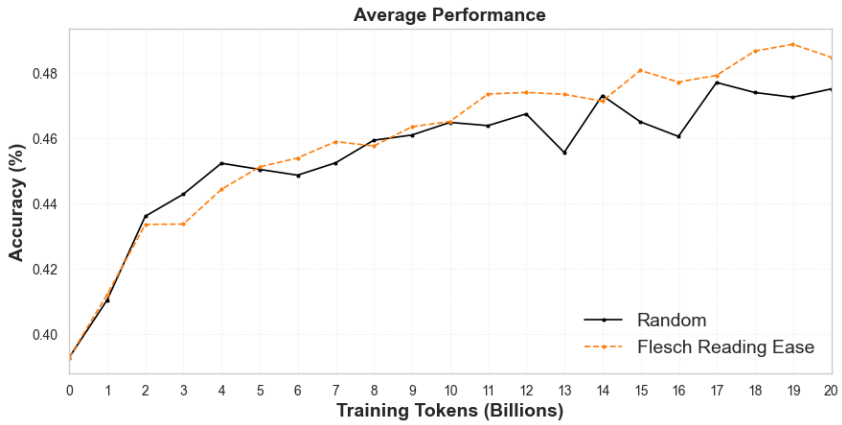

Figure 12: Performance of using Flesch Reading Ease with quadratic pacing, the model shows stable, non-convergent performance improvements across the full training horizon.

As shown in Figure 12, we observed again a stable performance gain till the end of the training for Flesch Reading Ease with quadratic pacing setting, and still no sign of convergence. The model trained on this setting achieves 2.5% higher accuracy, showing again the potential of this difficulty metric.

G.3 Curriculum Learning as Warmup

We are also interested in applying CL under S3 that use CL as a warmup phase of training, which previously showed impressive results.

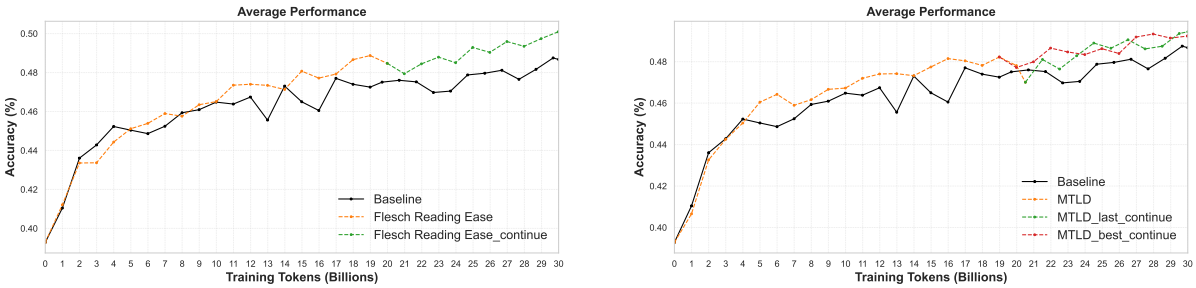

Figure 13: The average accuracy of models continually trained on two curriculum learning settings: Flesch Reading Ease with quadratic pacing (left) and MTLD with vanilla CL (right), the models were trained using 20B tokens for CL warmup, then we select the best checkpoint and the last checkpoints (they can be the same) from the CL settings to start continual training for the next 10B tokens.

We tested two settings: Flesch Reading Ease with quadratic pacing and MTLD with vanilla CL to serve as CL warmup pretraining with 20B tokens, then we continue training the model from either the best or the last chekcpoint for 10B tokens which are randomly shuffled. Our results is shown in Figure 13, where we observe same pattern as the 0.5B model: CL warmup lead the model achieves higher performance throughout the training, especially the continual training phase. Specifically, the Flesch Reading Ease setting and the MTLD setting achieved 3.2% and 2.1% higher peak performance compared to the baseline.

We then tested a 3B model on the MTLD setting with the same setup. We first train the model under curriculum learning for 20B tokens, then perform random training with 20B more tokens.

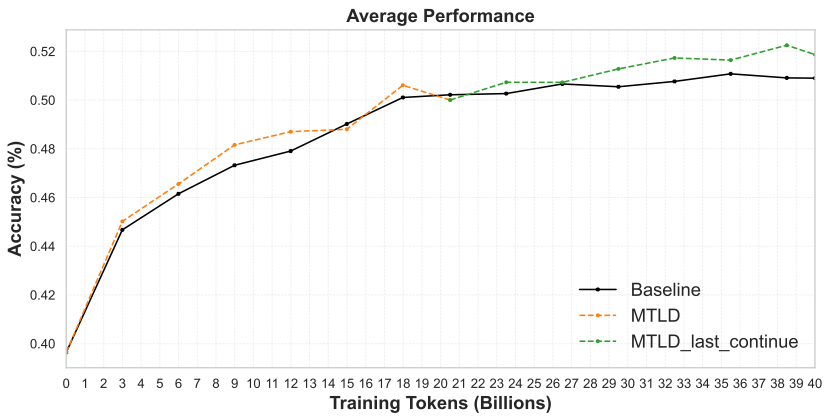

Figure 14: The average accuracy of model continually trained on MTLD with vanilla CL, the model is trained using 20B tokens for CL warmup, then we select the last checkpoint from the CL settings to start continual training for the next 20B tokens.

For the 3B model, we observed the same finding as in 1B and 0.5B models - CL warmup consistently improves performance, achieving a 2.0% gain over the baseline. These findings showed the scalability of CL as warmup setting, where the model can achieve consistent performance gains compared to the random baseline.

H Computation overhead

One important aspect of designing efficient curriculum learning settings is the computation overhead of computing the difficulty metrics. We show the gains in our experiments come without sustained additional

cost—CL involves only restructuring the initial data phase, making it a practical and low-overhead approach. We provide Table 2 showing time (rounded to 0.5 min) and cost estimates on a 72 vCPU AWS machine for computing difficulty metrics over 10B tokens. Additional cost for data loading and data ordering is 2$. While training a 0.5B model with 10B tokens requires 3e19FLOPs, accounting for about 84 GPU hours of A100. The computation overhead for calculating data difficulty and ordering data is thus negligible compared to pretraining.

Metric	**Copression Ratio**	**Fertility**	**Readability(FRE)**	**MTLD**	**#Tokens**	**Perplexity**
Time(mins)/cost($)	11.5/0.7	29.4/1.8	21/1.3	23.6/1.4	19.5/1.2	13/0.8

Table 2: Computation time and cost for the difficulty metrics.