Maximize Your Data's Potential: Enhancing LLM Accuracy with Two-Phase Pretraining

Anonymous ACL submission

Abstract

Pretraining large language models effectively requires strategic data selection, blending and ordering. However, key details about data mixtures especially their scalability to longer token horizons and larger model sizes remain underexplored due to limited disclosure by model developers. To address this, we formalize the concept of *two-phase pretraining* and conduct an extensive systematic study on how to select and mix data to maximize model accuracies for the two phases. Our findings illustrate that a two-phase approach for pretraining outperforms random data ordering and natural distribution of tokens by 3.4% and 17% on average accuracies. We provide in-depth guidance on crafting optimal blends based on quality of the data source and the number of epochs to be seen. We propose to design blends using downsampled data at a smaller scale of 1T tokens and then demonstrate effective scaling of our approach to larger token horizon of 15T tokens and larger model size of 25B model size. These insights provide a series of steps practitioners can follow to design and scale their data blends.

1 Introduction

Large language models (LLM) are typically pretrained on large amounts of data in the order of billions (B) or trillions (T) of tokens derived from multiple data sources such as web crawl, books, papers, patents, mathematical and legal documents, and so forth (Brown et al., 2020; Parmar et al., 2024b; Team et al., 2024b; Dubey et al., 2024a; Nvidia et al., 2024). To develop a state-of-the-art model, it is critical to understand the nature of these data sources and to make informed decisions about optimal data blending (how different data sources are weighed during pretraining) and training strategies. These decisions typically involve running multiple large-scale experiments to empirically investigate the optimal training data blend(s) and ordering of data.

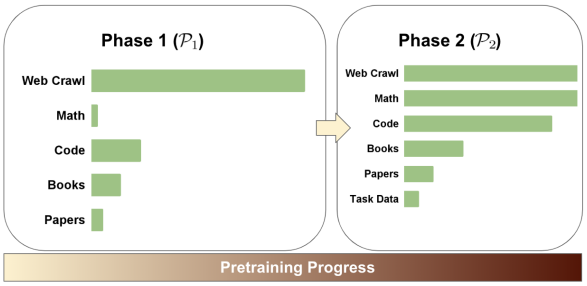

Figure 1: Diagram of our two phase training pipeline. Phase-1 blend encourages data diversity and phase-2 blend is focused on high quality datasets.

Most advanced models (OpenAI et al., 2024; Dubey et al., 2024b) do not divulge information on the data blends that are used, nor the ablation studies informing the data mixing and ordering decisions. Recent works (Blakeney et al., 2024; Groeneveld et al., 2024; Dubey et al., 2024b; Snowflake, 2024) provide high-level data blend information about a small portion of pretraining by encouraging the upsampling of certain domains towards the end. In general, there exists a knowledge gap regarding how to craft and choose an optimal data blend(s) for the entire training process, and the generalizability of data blends and ordering strategies to larger token horizons and model sizes.

In this work, we address the above knowledge gap by understanding optimal data blends and ordering strategies for training LLM. We formalize and extensively explore a two-phase training approach (Figure 1) that balances diversity and quality: phase-1 emphasizes diverse, high-quality web crawl data, while phase-2 focuses on high-quality data sources such as math, code, and wiki data. Specifically, in this work we propose to use downsampled data to prototype and explore multiple blends at a smaller scale of 1T tokens. We craft our blends based on quality of the data source and the number of epochs to be seen during pretraining. We then demonstrate the effectiveness of our

approach at a 15T token scale using the full data.

We evaluated on a comprehensive set of downstream tasking covering knowledge, reasoning, coding and math benchmarks. Our experiments illustrate that a quality and epoch based blend is better than a blend based on natural distribution by 13.2% and the two-phase approach is better than random ordering of data (blend is based on quality and epochs) by an average of 3.4% across downstream tasks. Furthermore, our results on downsampled data generalize across longer 15T token horizons on full data and larger model sizes, demonstrating the scalability and robustness of the two-phase approach. We also provide a fine-grained quality analysis of web crawl data, revealing optimal blending strategies to balance diversity and quality.

We share and highlight a series of findings made to create blends and order in our two-phase approach. Our main contributions are:

- Formalization and large-scale evaluation of the two-phase training approach for LLMs, with actionable strategies that enable effective LLM pretraining.
- Improving the understanding of data selection and blending with quality-based and epoch-based analyses of data, including web crawl.
- Demonstration of the scalability of blends using downsampled data at 1T to using full data at 15T tokens and larger model size of 25B.

2 A Two-Phase Approach to Pretraining

In this work, we explore a two-phased approach to pretraining: phase-1 ($\mathcal{P}_1$) then phase-2 ($\mathcal{P}_2$). Figure 1 demonstrates our two-phased approach. In each phase, we explore different data blends based on the quality and number of epochs to be seen of a data source. In phase-1 ($\mathcal{P}_1$), we explore a general data distribution which consists of a mix of web crawl data, medium-quality data, and low amounts of high-quality data. In phase-2 ($\mathcal{P}_2$), we explore a blend which includes task data and emphasizes high-quality datasets such as math, code, and high-quality web crawl (§5.1). As seen in Figure 1, our model sees the first general data blend during $\mathcal{P}_1$ for the majority of training, then a different data blend focused on high quality data during the shorter $\mathcal{P}_2$ of training.

The steps to create blends for $\mathcal{P}_1$ and $\mathcal{P}_2$ are: 1) Downsample a data source by a factor of f, 2) Estimate the quality of a data source (§5.1), 3) Estimate the epochs to be seen in the whole pretraining (§5.2) and finally 4) distribute the epochs appropriately in $\mathcal{P}_1$ and $\mathcal{P}_2$ (§3.2). The downsampling factor f is based on the final total token budget which we assume to be 15T similar to Dubey et al. (2024b). Hence, for us $f = 1/15$ i.e for each data source, the number of tokens available for pretraining is $1/15^{th}$ of the total token in that dataset. Downsampling helps to observe the impact of epochs of datasets at a smaller scale of 1T tokens and then can be used to scale the blend to a longer token horizon of 15T tokens using the full data.

Data Domain	Tokens (B)
Web Crawl	6244.3
Math	161.5
Wiki	16.7
Code	760.3
Books	776.3
Papers	212.6
$\texttt{CC}_{\texttt{dv}}$	348.3
Multilingual	1457.2
Task Data	6.6

Table 1: Tokens (billions) in each data domain.

Baselines: Since our blends are based on quality and epoch based analyses of the data as well as the ordering of the data in the two phases, we consider the following two baselines: 1) Natural Distribution Blend (BASE-ND): This blend is based on ratio of the number of tokens available in each data source. The weight for each dataset is equal to the total number of tokens in that dataset divided by the sum of tokens available in all the datasets. This weighting is neither based on quality nor the epochs to be seen for the dataset. 2) Random Order Pretraining (BASE-RO): This blend is based on quality and epochs of each dataset but does not use two phases to train the model. The weight for each dataset here is the same as our two-phase approach but the order in which the the dataset is seen during pretraining is random.

3 Experimental Setup

3.1 Data Sources

Our pretraining corpus spans a vast range of text data sources that cover several domains, types of data, and languages. We broadly divide our datasets into the following categories and their token counts in billions is shown in Table 1.

- **Web Crawl:** Data derived from Common Crawl (CC). We discuss the quality of this data and how to blend it in detail in §5.1.

Category	Domain	Blend1	Blend2	Blend3	Blend4	Blend5
Web Crawl	-	65.0	65.0	58.0	59.0	70.0
High Quality	Math	1.9	1.9	1.9	2.9	1.9
	Wiki	0.1	0.1	0.1	0.1	0.1
	Code	15.0	8.0	15.0	20.0	13.0
Medium Quality	Books	5.5	9.0	9.0	5.5	4.5
	Papers	3.5	5.0	5.0	3.5	1.9
	$\texttt{CC}_{\texttt{dv}}$	4.0	6.0	6.0	4.0	3.6
Multilingual	-	5.0	5.0	5.0	5.0	5.0

Table 2: Phase-1 Blends (in %)

Category	Domain	Blend1	Blend2	Blend3	Blend4	Blend5
Web Crawl	-	31.0	35.0	31.0	40.0	35.0
High Quality	Math	24.0	24.0	24.0	24.0	29.0
	Wiki	1.0	1.0	1.0	1.0	1.0
	Code	20.0	25.0	29.0	20.0	20.0
Medium Quality	Books	8.0	4.0	4.0	4.0	4.0
	Papers	4.0	2.0	2.0	2.0	2.0
	$\texttt{CC}_{\texttt{dv}}$	7.0	4.0	4.0	4.0	4.0
Multilingual	-	3.7	3.7	3.7	3.7	3.7
Task Data	-	1.3	1.3	1.3	1.3	1.3

Table 3: Phase-2 Blends (in %)

- **High-Quality:** This includes datasets from more specialized and professional domains such as mathematics (Paster et al., 2024; Stack Exchange, Accessed 2024), code (Li et al., 2023), and Wikipedia (wiki) data.
- **Medium-Quality:** Data derived from books & patents, papers (Gao et al., 2020), and Common Crawl derivatives ($\texttt{CC}_{\texttt{dv}}$) such as OpenWebText (Gokaslan and Cohen, 2019), BigScience (Laurençon et al., 2022), Reddit (Baumgartner et al., 2020), and CC-News. This category was determined by comparing this data to medium-quality crawl (see §5.1).
- **Multilingual:** Multilingual data (9 languages) derived from Wikipedia and Common Crawl.
- **Task Data:** This includes data used for supervised finetuning (SFT) during the alignment phase (Toshniwal et al., 2024; Nvidia et al., 2024). We also include the FLAN collection (Longpre et al., 2023).

3.2 Data Blends for Each Phase

The final blends in $\mathcal{P}_1$ and $\mathcal{P}_2$ are based on quality and epoch based ablations shown in §5.1 and §5.2. The insights from these studies are incorporated in Table 2 and 3.

In $\mathcal{P}_1$, we encourage diversity in data by including a high percentage of web crawl data which consists of high, medium, and low-quality crawl. We want to introduce a limited amount of high-quality data such as math, code, and wiki in $\mathcal{P}_1$. In $\mathcal{P}_2$, the emphasis is primarily on high-quality datasets and only includes a limited amount of medium-quality data. For example, in $\mathcal{P}_2$, we only use high-quality crawl instead of medium or low-quality (see §5.1).

Table 2 details the five blends explored in $\mathcal{P}_1$. These blends are designed to compare the proportion of high-level categories with each other. The difference between `Blend1` and `Blend2` is that `Blend2` has less code and more medium-quality datasets compared to `Blend1`. `Blend3` has less web crawl and more medium-quality datasets compared to `Blend1`. `Blend4` has less web crawl and more high-quality datasets compared to `Blend1`. `Blend5` is designed to have majority web crawl at the cost of code and medium-quality data.

Table 3 outlines the five blends explored in $\mathcal{P}_2$. In $\mathcal{P}_2$, we use more epochs and higher proportions of high-quality data such as high-quality web crawl, math, wiki, and code data. `Blend3` has more code and less medium-quality datasets compared to `Blend1`, and `Blend4` has more high-quality web crawl and less medium-quality datasets compared to `Blend1`. `Blend2` has a more balanced distribution among the data categories, while `Blend5` upsamples math data more heavily.

3.3 Model Specifications

We experiment using the Megatron (Shoeybi et al., 2020) model, an autoregressive causal left-to-right LLM, with the Tiktokenizer (OpenAI, 2023). We downsample all our data by factor $f = 1/15$. Hence, only $1/15$ of the tokens shown in Table 1 will be available for pretraining. We perform all our investigations using an 8 billion parameter model trained on 1 trillion total tokens. Furthermore, we test our two-phase approach by scaling along two dimensions: (1) we scale the token horizon to 1.7T tokens on a 8B model, and (2) we scale the parameters of the model to 25B and train on 1T tokens. Additionally, we train a 8B model on 15T tokens on full data (not downsampled) to observe if decisions made with downsampled data scales. Specifics on model architecture and hyperparameters are shared in Appendix A.

3.4 Evaluation Suite

To comprehensively assess our models, we use various benchmarks that evaluate different capabilities. These can be broadly divided into the following 4 categories, of which we report the final averages. We assess 5-shot accuracy for MMLU (Hendrycks

Exp.	MMLU	Reason.	GSM8K	Code	Avg.
BASE-ND	49.78	56.48	19.64	24.96	45.17
BASE-RO	56.49	59.69	30.86	35.55	51.12
Two-Phase	56.28	60.34	40.33	38.33	**52.86**

Table 4: Comparison of our two-phase training approach with BASE-ND and BASE-RO.

Tok.	MMLU	Reason.	GSM8K	Code	Avg.
1T	56.28	60.34	40.33	38.33	52.86
15T	70.30	64.11	64.82	46.38	59.84

Table 5: Results of our two-phase training approach with $\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend1 for downsampled data at 1T and then complete data at 15T.

et al., 2021), 0-shot accuracy[1] for reasoning tasks: CommonsenseQA (Talmor et al., 2019), ARC-Easy & Challenge (Clark et al., 2018), PIQA (Bisk et al., 2019), WinoGrande (Sakaguchi et al., 2019), HellaSwag (Zellers et al., 2019), OpenBookQA (Mihaylov et al., 2018), RACE (Lai et al., 2017), 0-shot accuracy for code benchmarks: HumanEval (+) (Chen et al., 2021) and MBPP (+) (Austin et al., 2021), and 8-shot chain-of-thought (CoT) accuracy for GSM8K (Cobbe et al., 2021). We also report a final overall *Avg.* for most results, which is an average over all individual evaluation tasks.

4 Results for Two-Phase Pretraining

Findings

- A two-phase approach for pretraining is effective.
- Phase-1 should focus on data diversity and phase-2 on high-quality data.

We compare our best blends $\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend1[2] using two-phase training with two baselines: 1) BASE-ND: the weights are determined by the tokens available in each dataset and are not based on quality, and 2) BASE-RO: the weights for all the datasets are the same in this and $\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend1. The only difference is the order in which the data is presented during training (random or two-phased). Table 4 illustrates that using a quality and epoch based blend is on average 13.2% better than natural distribution blend (compare BASE-RO vs BASE-ND) across downstream tasks. It also presents that using our two-phase training approach noticeably improves average accuracy by 3.4% compared to BASE-RO and 17% compared to BASE-ND. This empirically demonstrates that the strategy of two-phase training is useful and tasks such as code and math are sensitive to the ordering of high-quality data in the second phase.

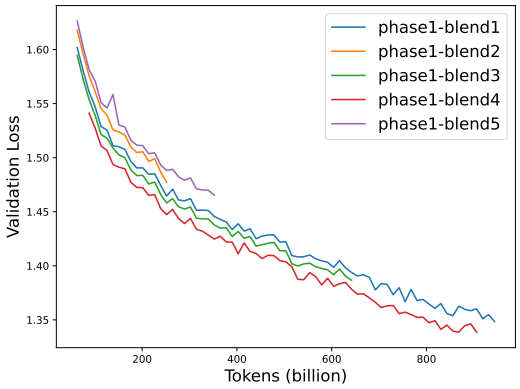

Figure 2: Phase-1 validation loss for different $\mathcal{P}_1$ blends.

We scale our best blend $\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend1 to 15T tokens and use the full dataset to train a 8B model. All the previous experiments are performed on downsampled data and 1T scale. This means that the number of epochs is constant in both the runs. Table 5 shows that blends crafted at smaller scale can generalize to longer token budgets if the quality and epochs of the datasets are maintained at scale. This shows the generalizability of our two-phase approach to pretraining as well as quality- and epoch-based approach to designing blends.

4.1 Determining Blends

As discussed in §3.2, we explore five different blends for phase-1.[3] We train an 8B model on downsampled data for 1T tokens for all five blends and eliminate blends based on a separately held-out validation split. Fig. 2 illustrates the validation loss for all five blends. As we can see, Blend5 and Blend2 had 2.8% and 2.1% higher validation loss, respectively, relative to Blend4 at approx. 250B tokens. Hence, we discontinue these two blends at that point. Since, the validation loss of the remaining three blends was within a margin of 1%, we periodically evaluate their accuracy on downstream

[1] We use normalized accuracy for ARC-Easy, ARC-Challenge, PIQA, HellaSwag, and OpenBookQA.

[2] see §4.1 Tab. 7 on how we select this blend and §5.3 Tab. 14 for the duration of $\mathcal{P}_2$ for best results.

[3] More detailed evaluation results of the major experiments in this section broken down by individual reasoning, MMLU, and code benchmarks and categories can be found in §B.

Exp.	Tokens	MMLU	Reason.	GSM8K	Code	Avg.
$\mathcal{P}_1$-Blend1	200	34.72	52.83	6.14	16.43	38.81
$\mathcal{P}_1$-Blend3	200	36.78	51.97	6.22	15.18	38.10
$\mathcal{P}_1$-Blend4	200	38.70	53.81	11.30	18.21	**40.48**
$\mathcal{P}_1$-Blend1	250	42.51	54.52	7.51	16.14	40.35
$\mathcal{P}_1$-Blend3	250	40.41	53.87	8.11	15.62	39.72
$\mathcal{P}_1$-Blend4	250	42.76	54.99	10.16	19.29	**41.66**
$\mathcal{P}_1$-Blend1	629	51.93	57.83	14.94	22.97	45.28
$\mathcal{P}_1$-Blend3	629	52.44	57.74	15.39	22.43	45.15
$\mathcal{P}_1$-Blend4	629	52.78	58.11	18.27	24.24	**46.07**

Table 6: $\mathcal{P}_1$ results after various token counts (billions).

Exp.	MMLU	Reason.	GSM8K	Code	Avg.
$\mathcal{P}_1$-Blend1	55.00	59.12	20.09	23.56	46.76
$\mathcal{P}_1$-Blend4	56.25	59.54	23.43	27.61	48.40
$\mathcal{P}_1$-Blend1-$\mathcal{P}_2$-Blend1	56.04	60.04	37.00	36.19	51.88
$\mathcal{P}_1$-Blend1-$\mathcal{P}_2$-Blend2	55.88	60.15	36.85	35.89	51.84
$\mathcal{P}_1$-Blend1-$\mathcal{P}_2$-Blend3	55.80	60.08	39.80	35.75	51.96
$\mathcal{P}_1$-Blend1-$\mathcal{P}_2$-Blend4	56.15	60.26	36.85	36.30	51.88
$\mathcal{P}_1$-Blend1-$\mathcal{P}_2$-Blend5	56.49	60.41	36.92	34.40	51.65
$\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend1	56.58	60.18	37.98	37.01	**52.28**
$\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend2	56.89	60.00	36.62	36.97	52.10
$\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend3	56.10	60.08	39.27	35.15	51.78
$\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend4	57.03	59.98	36.85	36.30	51.93
$\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend5	56.92	60.35	38.29	34.56	51.77

Table 7: Evaluation results after $\mathcal{P}_2$ of training.

tasks. Table 6 shows the results of the remaining three phase-1 blends at various token counts. At each token evaluation point – 200B, 250B and 629B, we see that Blend3 is consistently worse than the other two blends. Hence, we eliminate this blend after 629B tokens of training. For this experiment, we switch from $\mathcal{P}_1$ to $\mathcal{P}_2$ after ≈70% of training, i.e the last 30% of training is $\mathcal{P}_2$. In §5.3, we explore varying the percentage of $\mathcal{P}_2$.

Results in Table 6 follow intuition since Blend4 has the highest amount of high-quality data and is hence better than Blend1 and Blend3. Blend3 has more medium-quality data at the cost of web crawl compared to Blend1. This result confirms that books, papers, and CC_{dv} are of medium-quality compared to our high-quality datasets and our web crawl blend.

Finally, we explore five different blends of $\mathcal{P}_2$ described in Table 3 in combination with $\mathcal{P}_1$-Blend1 and $\mathcal{P}_1$-Blend4. Hence, we have ten different combinations of $\mathcal{P}_1$ and $\mathcal{P}_2$ blends. Table 7 shows the results on all ten combinations of blends. We find that $\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend1 performs the best on average. Table 7 also presents the final results of $\mathcal{P}_1$-Blend1 and $\mathcal{P}_1$-Blend4 if only the $\mathcal{P}_1$ blend was continued for 1T tokens without ever switching

Exp.	Tok.	MMLU	Reason.	GSM8K	Code	Avg.
$\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend1	1T	56.58	60.18	37.98	37.01	52.28
$\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend1	1.7T	56.61	60.88	42.15	37.62	53.28
$\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend6	1.7T	59.85	61.63	43.90	39.61	**54.45**

Table 8: Scaling results for 1.7T tokens vs. 1T tokens, with and without high-quality data epoch adjustment.

to $\mathcal{P}_2$ blends. It shows that switching to any of the $\mathcal{P}_2$ blends for training is better than continuing the $\mathcal{P}_1$ blends for all metrics. We observe the largest absolute gains in GSM8K and code of 14.6% and 9.4%, respectively, for $\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend1.

4.2 Scaling

Findings

- Two-phase approach is scalable and robust to token horizon and model scale.
- Data blends need adjusting at longer token horizons based on epoch count to avoid high-quality data overexposure.

We further explore scaling our best blend along two dimensions: (1) a longer token horizon of 1.7 trillion tokens and (2) larger model size of 25B parameters. For a longer token horizon, we aim to assess whether the blend can be used as is or if adjustments are necessary to prevent overfitting (observed in §5.2). Note that this is different from scaling to 15T token where we use the full data. Here we still use the downsampled data and scale to 1.7T tokens and hence the epochs seen of each dataset would be higher. Since high number of epochs of high-quality datasets are primarily seen in $\mathcal{P}_2$ of pretraining, we create a new blend, $\mathcal{P}_2$-Blend6[4], which is an epoch-adjusted version of $\mathcal{P}_2$-Blend1 to ensure that we do not see more than 8 epochs of certain high-quality data sources like math and task data. Table 8 shows the comparison of scaling from 1T to 1.7T total tokens. We see that $\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend6 is on average 2.2% better than $\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend1, illustrating that we need to adjust our blends according to the epoch counts of high-quality data for optimal results. Both the 1.7T models are better than 1T, demonstrating that we can still obtain higher downstream accuracies by training on more tokens, even if it means training on more than 8 epochs of high-quality data.

We also investigate if our best blend can scale to a larger model size. Given the high number

[4]We show comparison of Blend1 and Blend6 in Table 15.

Model Size	MMLU	Reason.	GSM8K	Code	Avg.
8B	57.31	61.16	45.11	38.97	53.92
25B	65.97	63.29	59.14	45.57	**58.47**

Table 9: Evaluation results for 8B vs. 25B parameter models, using the same blend: $\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend1. Note that we use a maximum sequence length of 8192 (instead of 4096) for both models here.

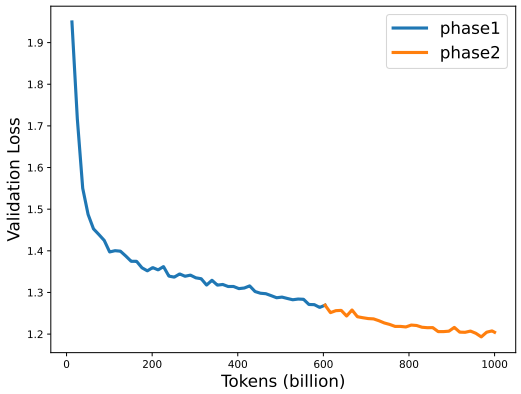

Figure 3: Validation loss for the 25B model using two-phase training with $\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend1.

of epochs of high-quality data in $\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend1, we also want to determine if a model with a larger capacity might memorize the data and overfit on it. Figure 3 shows that the validation loss is always decreasing for the 25B model, indicating that there is no overfitting with $\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend1. Table 9 shows results for the 25B model compared to the 8B model on this data blend combination. Understandably, the 25B model is substantially better across the board, demonstrating that the two-phase training approach and data blend combination can also scale to larger model sizes.

5 Ablations

This section details quality-based data blending, epoch study and percentage of phase-2 to be conducted in the pretraining. Additional fine-grained analyses of data blends and study on learning rate schedule to be used in phase-2 is shown in Appendix §D and §E.

5.1 Quality-Based Data Blending

Insights

- Upsampling high quality and not using low quality CC data is most effective.

Quality Label	Token	CC-Blend1	CC-Blend2	CC-Blend3	CC-Blend4
High	35.96%	57.0	57.0	51.5	45.0
Medium-High	8.56%	25.0	25.0	23.5	20.0
Medium	34.25%	18.0	13.0	23.0	32.0
Medium-Low	15.41%	0.0	2.0		
Low	5.82%	0.0	3.0	2.0	3.0

Table 10: CC blends (in %) by quality. For CC-Blend3 and CC-Blend4, we merged the Medium and Medium-Low categories. *Token* column refers to the the natural distribution of tokens, i.e. percentage of total CC data that belongs to each category.

Exp.	MMLU	Reason.	GSM8K	Code	Avg.
CC-Blend1	57.09	61.16	13.42	19.78	46.01
CC-Blend2	56.69	61.77	14.18	19.56	45.11
CC-Blend3	56.29	60.74	14.25	18.44	44.17
CC-Blend4	55.73	60.57	14.31	18.50	44.06

Table 11: $\mathcal{P}_1$ results using our various CC blends.

- CC_{dv}, papers and books are similar in quality to CC-Medium-High.

The data blends of our two-phase approach are mainly based on the assessment of each data source's quality. Hence, we carry out extensive experiments to find an optimal data blend for web crawl documents. While previous work (Dubey et al., 2024a; Yang et al., 2024; Team et al., 2024a) mentions that web crawl documents like Common Crawl (CC) form a large majority of their pretraining data, none of them share a recipe on how to mix different slices of CC. Some recent work on constructing crawl-based pretraining datasets (Penedo et al., 2024b; Li et al., 2024a) directly use the high quality crawl documents in pretraining but provide no specific data mixing strategy. In this section, we provide comprehensive details on how to create a data blend for CC documents and use it effectively in our phase-1 and phase-2 of pretraining. Additionally, we provide a quality assessment of other datasets like CC_{dv}, papers, books and our high quality datasets. We compare them with medium, and high quality web crawl to position them optimally in our $\mathcal{P}_1$ and $\mathcal{P}_2$ blends.

Quality-Based Blending for Web Crawl: Each document in our web crawl data is classified into one of five quality categories: High, Medium-High, Medium, Medium-Low, and Low using the classifier from Su et al. (2024).

We investigate various blends using quality-based weighted sampling approach[5] for all of web

[5]We show comparison of quality-based blending with nat-

Dataset	MMLU	Reason.	GSM8K	Avg.
CC-Medium	52.86	56.00	18.04	42.30
CC-Medium-High	53.75	58.09	18.50	43.45
CC-High	55.82	59.65	20.85	45.44
High Quality	54.53	58.51	24.11	45.72
CC_{dv}	54.47	58.20	20.14	44.27
Books	55.36	58.93	18.50	44.26
Papers	54.55	58.65	19.41	44.20

Table 12: Results of different quality crawl and their comparison with other datasets. Since code data is not included in most of these experiments, we exclude code evaluation.

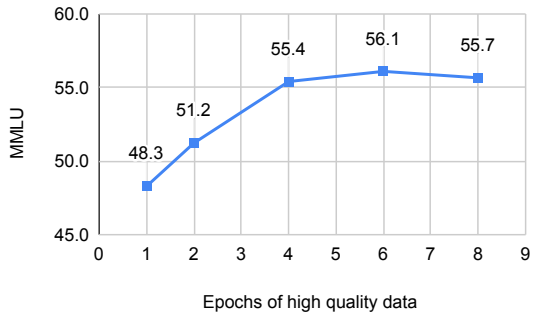

Figure 4: MMLU accuracy (%) vs. number of epochs of high-quality crawl in the data mix.

crawl data from 99 CC snapshots for our phase-1 of pretraining. The idea is to upsample high and medium-quality crawl documents while avoiding a high quantity of low-quality data. The overall idea for the four web crawl blends in Table 10 is to iteratively decrease the percentage of tokens from High and Medium-High and increase the tokens in the lower categories. The results in Table 11[6] demonstrate that eliminating the tail-end of the web crawl data belonging to Medium-Low and Low quality categories is beneficial as opposed to to keeping them for diversity. Based on these results, we choose CC-Blend1 as the final data blend for web crawl documents to be used in all our final $\mathcal{P}_1$ blends (§4 and Table 2). For $\mathcal{P}_2$, we only use web crawl data that belongs to *High*-quality category.

Quality Estimation of Other Datasets: We assess how our CC_{dv}, papers, books and high quality datasets such as math, code and Wiki compare to CC-Medium,CC-Medium-High and CC-High quality crawl data. We continue training the last checkpoint of $\mathcal{P}_1$-Blend4, for an additional 50B tokens using a data mix that consisted 66% of the data being tested, mixed with 34% of CC-High.

The results in Table 12 show that CC_{dv}, papers and books datasets have similar accuracies to CC-Medium-High on the majority of benchmarks, and lag behind CC-High. As such, we group them under the "medium-quality" data category for our experiments (see §3.1). The high quality datasets have an average accuracy better than CC-High.

ural token distribution-based blend in §C.

[6]The average in this table is primarily based on reasoning tasks and MMLU because these blends do not have math or code data.

Domain	Epochs	MMLU	Reason.	GSM8K	Code	Avg.
Math	1	57.06	60.51	36.47	33.55	51.49
Math	4	57.01	60.32	38.21	35.20	51.92
Math	8	56.58	60.18	37.98	37.01	**52.28**
Math	12	56.09	59.69	38.29	35.70	51.63
Task Data	1	56.37	59.39	34.50	30.57	49.84
Task Data	4	56.57	59.46	40.18	35.27	51.53
Task Data	8	56.58	60.18	37.98	37.01	**52.28**
Task Data	12	56.77	59.96	38.44	36.04	51.93

Table 13: Results of varying the number of epochs of math and task data during $\mathcal{P}_2$ of training.

5.2 Epoch-Based Analysis

Insights

- We recommend 6 epochs of high-quality crawl and 8 epochs of math and task data for data mixing.

We take the number of epochs of high quality datasets into account while creating our $\mathcal{P}_1$ and $\mathcal{P}_2$ blends. We experiment with different numbers of epochs for high-quality crawl, math, and task data.

Since, majority of web crawl is used in $\mathcal{P}_1$, we pretrained an 8B model with 1T tokens, using different epochs of high-quality crawl tokens in the data mix, and evaluate each model's MMLU score. Note that we keep the overall percentage of web crawl the same in all the experiments. As we can see in Figure 4, increasing the number of high-quality tokens increases the MMLU score until 6 epochs. We primarily present MMLU score because these experiments do not include high amount of math or code data.

Since, majority of math and task-data is seen in $\mathcal{P}_2$, Table 13 presents results for different numbers of epochs for them in $\mathcal{P}_2$. It shows that ≈ 8 epochs of math is a good balance while not sacrificing accuracy on MMLU and reasoning. For task data, all metrics generally improve with more epochs, although there appears to be diminishing returns on several past epoch 8. Note that 8 epochs of

$\mathcal{P}_2$ %	MMLU	Reason.	GSM8K	Code	Avg.
0	56.10	61.81	16.60	16.22	37.68
10	56.52	59.70	33.13	32.55	50.48
20	56.54	59.93	40.16	34.29	51.58
30	56.58	60.18	37.98	37.01	52.28
40	56.28	60.34	40.33	38.33	**52.86**
50	55.94	59.82	37.68	36.86	51.96

Table 14: Results of different durations of $\mathcal{P}_2$ using $\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend1.

both math and task data corresponds to our best $\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend1 blend combo from §4.

5.3 Optimal Duration of Phase-2

Insights

- Pretraining with the $\mathcal{P}_2$ blend for the final 40% gives the best results.

We investigate the percentage of phase-2 to use in the whole pretraining regime. We experiment with 0 to 50% of $\mathcal{P}_2$ in the whole of pretraining. The longer the duration of $\mathcal{P}_2$, the shorter the duration of $\mathcal{P}_1$. We use the $\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend1 blend combination that we found best in §4.

Table 14 illustrates that a higher percentage of $\mathcal{P}_2$ until 40% is better overall, especially in math and code. Going above this, e.g. to $\mathcal{P}_2$ as 50% of training, downstream accuracies start to degrade across the board, potentially due to overfitting.

6 Related Work

Selecting and structuring pretraining datasets is important to improve model generalization and efficiency. Dubey et al. (2024b) emphasize openness and accessibility of models, Computer (2023); Soldaini et al. (2024) assemble an open corpus of trillions of tokens for large-scale training, and Groeneveld et al. (2024) release a truly Open Language Model, including its framework, training data, and code. Studies such as Li et al. (2024b); Penedo et al. (2024a) demonstrating that refined data selection impacts model accuracies more significantly than simply the quantity of data. But these studies are primarily aimed at CC data and they do not suggest any data mixing strategies for pretraining. Parmar et al. (2024a) provide a systematic approach to building effective LLM pretraining datasets with ablations on data attributes, and existing curation, selection, and sampling methods. In our work, we provide a systematic approach to craft data blends and to order the data in pretraining.

Strategic weighting and timing of data usage can also noticeably impact model accuracies. Techniques like domain upsampling (Blakeney et al., 2024; Dubey et al., 2024b) towards the end of training have been shown to be effective. Snowflake (2024); Groeneveld et al. (2024) provide details about high level blends for their pretraining process. In contrast, our work provides fine grained details about the data blend creation process along with actionable steps that model developers can use to develop data blends and order. Prior work (Shen et al., 2023; Longpre et al., 2024; Mindermann et al., 2022; Xie et al., 2023a,b; Shao et al., 2024) investigates optimizing data mixtures based on clustering methods, manually designed domain composition weights, proxy models or reference models to determine data composition weights and sample-level data selection. Our work primarily focuses on data ordering and scaling of data blends in pretraining and can be used in conjunction with other data sampling techniques.

Curriculum learning approaches inspired by human learning offer an ordered way to introduce data gradually to enhance model learning. Martinez et al. (2023); Wang et al. (2022); Feng et al. (2024) investigate cognitively-motivated curriculum-based training including vocabulary, and objective curricula, and outline and the challenges and potential solutions for designing effective curricula. Our work shows that ordering of data based on quality in pretraining LLMs has a significant impact of downstream accuracies.

7 Conclusion

In conclusion, through extensive experiments, we demonstrate the effectiveness of a two-phase pretraining approach for LLM. For the initial training phase, a more general data distribution consisting of mainly of web crawl proves most effective, while phase two benefits from a comprehensive data blend, with additional focus on math, code, and task data. Phase-two for the last ≈40% of training yields the best results, and over-extending it leads to diminishing returns. Increasing model size and token horizon further enhances accuracy, demonstrating the scalability of our approach. Importantly, we also show that considering both the quality of the data (including web crawl) and the number of epochs of each data source is crucial to attain optimal results and prevent overfitting.

Limitations

Some limitations of our work include our present suite of models and evaluation benchmarks. We can extend our work and show the effectiveness of two-phase pretraining approach on more LLM architectures such as Mamba (Gu and Dao, 2023), other hybrid SSM based architectures (Glorioso et al., 2024; Lieber et al., 2024) and mixture of experts (Shazeer et al., 2017). While our evaluation benchmarks are quite comprehensive, we could potentially expand to an even broader range of evaluations, including nuanced domain-specific or interactive tasks, or more theory of mind and developmental psychology-inspired benchmarks. This includes assessing capabilities such as analogical reasoning (Webb et al., 2023). Further, our scaling experiments could be expanded. Scaling up to hundreds of billions of parameters or significantly longer training may yield additional insights. Lastly, while our work focuses on two-phase training and shows its efficacy, we can potentially investigate multi-phase training, and the impact of the order of the phases. However, we believe this is more suited for future work. Overall, these are directions to potentially improve and expand upon our work. Despite these potential limitations, we feel that our current work is an insightful and useful contribution to the research community.

Ethical Considerations

Our research uses publicly available and commonly used datasets in LLM development. These sources, including Common Crawl, Wikipedia, and code repositories, are widely adopted in the research community. We examined the quality and origins of our data, prioritizing high-quality, domain-relevant data sources to improve LLM capabilities in a responsible manner. However, web crawl data may inherently contain biases or inappropriate content despite filtering efforts. We used established data cleaning and quality assurance procedures but acknowledge that potential biases may persist and impact model behavior in certain circumstances.

We recognize that scaling models and exploring data blending strategies require significant computational resources, which may raise environmental concerns. To mitigate this, we focused on efficient training strategies, such as two-phase training, to improve accuracy without excessively increasing resource usage. Future studies could benefit from exploring energy-efficient training methods to further minimize the environmental impact.

Our models, data blends, and accompanying publication are intended solely for research purposes, with no intended real-world application without additional safety evaluations. We caution against deploying models based on our methods without thorough testing, as they may carry unknown risks, particularly when applied to tasks involving sensitive or personal information. Our work aims to advance the understanding of LLM training strategies, and we feel that it is an important contribution to the research community. We encourage researchers to expand upon our work while further investigating the ethical and societal implications of LLM.

References

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. 2023. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901, Singapore. Association for Computational Linguistics.

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

Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. 2020. The pushshift reddit dataset. *Preprint*, arXiv:2001.08435.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2019. Piqa: Reasoning about physical commonsense in natural language. *Preprint*, arXiv:1911.11641.

Cody Blakeney, Mansheej Paul, Brett W. Larsen, Sean Owen, and Jonathan Frankle. 2024. Does your data spark joy? performance gains from domain upsampling at the end of training. *Preprint*, arXiv:2406.03476.

Tom B. Brown et al. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Mark Chen et al. 2021. Evaluating large language models trained on code. *Preprint*, arXiv:2107.03374.

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

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias

Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Together Computer. 2023. Redpajama: an open dataset for training large language models.

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

Abhimanyu Dubey et al. 2024b. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online. Association for Computational Linguistics.

Steven Y. Feng, Noah Goodman, and Michael Frank. 2024. Is child-directed speech effective training data for language models? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22055–22071, Miami, Florida, USA. Association for Computational Linguistics.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The pile: An 800gb dataset of diverse text for language modeling. *Preprint*, arXiv:2101.00027.

Paolo Glorioso, Quentin Anthony, Yury Tokpanov, James Whittington, Jonathan Pilault, Adam Ibrahim, and Beren Millidge. 2024. Zamba: A compact 7b ssm hybrid model. *arXiv preprint arXiv:2405.16712*.

Aaron Gokaslan and Vanya Cohen. 2019. Openwebtext corpus. `http://Skylion007.github.io/OpenWebTextCorpus`.

Dirk Groeneveld et al. 2024. OLMo: Accelerating the science of language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15789–15809, Bangkok, Thailand. Association for Computational Linguistics.

Albert Gu and Tri Dao. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization. *Preprint*, arXiv:1412.6980.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

Hugo Laurençon et al. 2022. The bigscience roots corpus: A 1.6tb composite multilingual dataset. In *Advances in Neural Information Processing Systems*, volume 35, pages 31809–31826. Curran Associates, Inc.

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

Jeffrey Li et al. 2024b. Datacomp-LM: In search of the next generation of training sets for language models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Raymond Li et al. 2023. Starcoder: may the source be with you! *Transactions on Machine Learning Research*. Reproducibility Certification.

Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, Itay Dalmedigos, Erez Safahi, Shaked Meirom, Yonatan Belinkov, Shai Shalev-Shwartz, et al. 2024. Jamba: A hybrid transformer-mamba language model. *arXiv preprint arXiv:2403.19887*.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. In *International Conference on Machine Learning*, pages 22631–22648. 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.

Richard Diehl Martinez, Hope McGovern, Zebulon Goriely, Christopher Davis, Andrew Caines, Paula Buttery, and Lisa Beinborn. 2023. CLIMB – curriculum learning for infant-inspired model building. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 112–127, Singapore. Association for Computational Linguistics.

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

Sören Mindermann, Jan M Brauner, Muhammed T Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N Gomez, Adrien Morisot, Sebastian Farquhar, and Yarin Gal. 2022. Prioritized training on points that are learnable, worth learning, and not yet learnt. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 15630–15649. PMLR.

Nvidia et al. 2024. Nemotron-4 340b technical report. *Preprint*, arXiv:2406.11704.

OpenAI. 2023. tiktoken. https://github.com/openai/tiktoken. Accessed: 2024-12-14.

OpenAI et al. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Jupinder Parmar, Shrimai Prabhumoye, Joseph Jennings, Bo Liu, Aastha Jhunjhunwala, Zhilin Wang, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. 2024a. Data, data everywhere: A guide for pretraining dataset construction. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10671–10695, Miami, Florida, USA. Association for Computational Linguistics.

Jupinder Parmar et al. 2024b. Nemotron-4 15b technical report. *Preprint*, arXiv:2402.16819.

Keiran Paster, Marco Dos Santos, Zhangir Azerbayev, and Jimmy Ba. 2024. Openwebmath: An open dataset of high-quality mathematical web text. In *The Twelfth International Conference on Learning Representations*.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. 2024a. The fineweb datasets: Decanting the web for the finest text data at scale. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

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

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande: An adversarial winograd schema challenge at scale. *Preprint*, arXiv:1907.10641.

Yunfan Shao, Linyang Li, Zhaoye Fei, Hang Yan, Dahua Lin, and Xipeng Qiu. 2024. Balanced data sampling for language model training with clustering. *arXiv preprint arXiv:2402.14526*.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.

Zhiqiang Shen, Tianhua Tao, Liqun Ma, Willie Neiswanger, Zhengzhong Liu, Hongyi Wang, Bowen Tan, Joel Hestness, Natalia Vassilieva, Daria Soboleva, et al. 2023. Slimpajama-dc: Understanding data combinations for llm training. *arXiv preprint arXiv:2309.10818*.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2020. Megatron-lm: Training multi-billion parameter language models using model parallelism. *Preprint*, arXiv:1909.08053.

Snowflake. 2024. Snowflake arctic: The best llm for enterprise ai — efficiently intelligent, truly open. https://www.snowflake.com/en/blog/arctic-open-efficient-foundation-language-\models-snowflake/. Accessed: 2024-12-14.

Luca Soldaini et al. 2024. Dolma: an open corpus of three trillion tokens for language model pretraining research. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15725–15788, Bangkok, Thailand. Association for Computational Linguistics.

Stack Exchange. Accessed 2024. Stack exchange data dump.

Dan Su, Kezhi Kong, Ying Lin, Joseph Jennings, Brandon Norick, Markus Kliegl, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Nemotron-cc: Transforming common crawl into a refined long-horizon pretraining dataset. *Preprint*, arXiv:2412.02595.

Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. 2021. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024a. Gemma 2:

Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Gemma Team et al. 2024b. Gemma: Open models based on gemini research and technology. *Preprint*, arXiv:2403.08295.

Shubham Toshniwal, Ivan Moshkov, Sean Narenthiran, Daria Gitman, Fei Jia, and Igor Gitman. 2024. Openmathinstruct-1: A 1.8 million math instruction tuning dataset. *arXiv preprint arXiv:2402.10176*.

Xin Wang, Yudong Chen, and Wenwu Zhu. 2022. A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):4555–4576.

Taylor Webb, Keith J. Holyoak, and Hongjing Lu. 2023. Emergent analogical reasoning in large language models. *Preprint*, arXiv:2212.09196.

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. 2023a. Doremi: Optimizing data mixtures speeds up language model pretraining. In *Advances in Neural Information Processing Systems*, volume 36, pages 69798–69818. Curran Associates, Inc.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy S Liang. 2023b. Data selection for language models via importance resampling. *Advances in Neural Information Processing Systems*, 36:34201–34227.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

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

Biao Zhang and Rico Sennrich. 2019. Root mean square layer normalization. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Category	Domain	Blend1	Blend6
Web Crawl	-	31.0	49.0
High Quality	Math	24.0	14.4
	Wiki	1.0	0.6
	Code	20.0	12.0
Medium Quality	Books	8.0	8.0
	Papers	4.0	4.0
	CC_{dv}	7.0	7.0
Multilingual	-	3.7	4.2
Task Data	-	1.3	0.8

Table 15: Comparison of `Blend1` and `Blend6` (in %) used for scaling the token budget in §4.2 and Table 8.

A Model Specifications

We use RoPE position embeddings (Su et al., 2021), RMSNorm layer normalization (Zhang and Sennrich, 2019), with Grouped Query Attention (Ainslie et al., 2023). The maximum sequence length is 4096. We use a global batch size of 1536, and the Adam optimizer (Kingma and Ba, 2017) with $\beta = (0.9, 0.95)$ and $\epsilon = 1e\text{-}08$.

$\mathcal{P}_1$ training uses cosine LR decay with an initial LR of $3e$-4 and targeted to reach a min-LR of $3e$-6 at the end of the full training run (both phases). We start $\mathcal{P}_2$ with the intermediate LR reached at the end of $\mathcal{P}_1$, and anneal using cosine LR decay to $3e$-6 (§E). Our experiments are run using up to 1024 NVIDIA H100 GPUs.

B Detailed Two-Phase Pretraining Results (Reasoning, MMLU, Code)

Tables 18 to 25 contain detailed evaluation results of the major experiments reported in §4 for reasoning, MMLU, and code, broken down by individual categories and benchmarks. They correspond to the results found in Tables 7 to 9 in §4.

Table 15 shows the comparison of `Blend1` and `Blend6` used in scaling experiments in Table 8. If we use the same `Blend1` as is and train for more number of tokens (1.7T) then the number of epochs seen of each dataset would be higher compared to 1T training.

C Details of Quality-Based Data Blend

We first compare a baseline blend (ND) which uses the natural distribution of tokens with a smartly constructed weighted sampling blend (WS). ND is based on the number of tokens that belong in each category as opposed to utilizing the quality label i.e. if 59% of the tokens belong to *Low* then 59% of tokens seen during pretraining would be-

Quality Label	ND	WS
High	0.01	0.04
Medium-High	1.08	6.42
Medium	7.01	41.83
Medium-Low	26.46	25.09
Low	64.44	0.00

Table 16: Data blends for CC Quality estimation experiment. The overall percentage of the Common Crawl Snapshots in our experiments is fixed at 73.3%.

Data Mixing	MMLU	Reason.	GSM8K	Code
ND	42.94	59.40	8.11	19.25
WS	56.10	61.60	11.98	17.41

Table 17: Our *WS*: weighted sampling data mixing method outperforms the *ND*: natural distribution method.

long to *Low*. We then create a data blend (WS) based on weighted sampling of high and medium-quality tokens. The idea is to upsample high and medium-quality crawl documents and not use the low-quality data at all. Table 16 shows the token percentages that belong to each of the five quality labels for both ND and WS blends. Table 17 illustrates the results of the two models trained on the ND and WS data blends of web crawl, respectively. We see that our data blend (WS) outperforms on most of the evaluation tasks by a large margin, and the improvement on MMLU is substantial.

D Finegrained $\mathcal{P}_2$ Blend Experiments

We investigate fine-grained $\mathcal{P}_2$ blends to determine the optimal blend. For these experiments, we use a model trained on a $\mathcal{P}_1$ blend for 900B tokens (10% $\mathcal{P}_2$ duration), with a linear LR decay to 0.

Crawl, Math, & Code: We investigate different percentages of high-quality crawl, math, and code data as shown in Table 26. Table 28 demonstrates that a higher amount of math data (i.e. 30%) helps across the board. However, code data results are mixed, as too much code without enough math (`CMC-B1`) seems to hurt all non-code metrics. Comparing (`CMC-B2`) vs. (`CMC-B3`), more than 15% code does not add as much value, as gains saturate. Trading off crawl data for more code data also slightly hurts MMLU. As such, we decide that a final blend consisting of a higher amount of crawl and math with a moderate amount of code seems best overall. This corresponds to `CMC-B3` in Table 26, which consists of 30% crawl, 33% math, and 15% code.

Task Data: Second, we investigate the inclusion of task data. Specifically, adding FLAN and synthetically-generated GSM8K-train data (similar to data augmentation approaches (Feng et al., 2021)) to the `CMC-B3` blend. Our FLAN data consists of a mixture of normal FLAN and FLAN-CoT (chain-of-thought) data. We compare 10 and 20 epochs of FLAN. These blends can be found in Table 27, with the results in Table 28. We can see that including synthetic GSM8K-train and FLAN data noticeably improves GSM8K scores while not detrimenting the other benchmarks. In fact, FLAN data also helps further improve MMLU and reasoning. 20 epochs of FLAN seems better than 10 epochs overall. Hence, including task data for $\mathcal{P}_2$ of training seems to be a good idea.

All Data Mixture: Lastly, we investigate a final $\mathcal{P}_2$ data mixture which is a combination of all the data sources we tried, including FLAN, GSM8K, and relatively higher amounts of math and code data. For this experiment, we use 30% upsampling with LR cosine decay to $3e-6$. This blend can be found in Table 27, with the results at the bottom of Table 28.[7] We find that mixing all data sources helps greatly with GSM8K, noticeably with coding and reasoning, while retaining accuracy on MMLU. Hence, the final $\mathcal{P}_2$ blends we investigate in §4 (Table 3) are motivated by these ablations – they are blends of all data sources, including task data, with higher proportions of math and code.

E Annealing Learning Rate Schedule

We investigate different learning rate (LR) schedules for phase $\mathcal{P}_2$. Specifically, using the same $\mathcal{P}_2$ blend for a 10% duration, we try different LR strategies. We compare cosine vs. linear LR decay functions, and also compare decaying to a final LR of 0 vs. $3e$-6 (1% of the original $\mathcal{P}_1$ starting LR of $3e$-4). Not decaying LR entirely to 0 leaves room for post-training, which is likely preferable.

As seen in Table 29, there is a negligible difference between linear and cosine LR decay, so we choose cosine decay for consistency with $\mathcal{P}_1$. We also see that LR decay to $3e$-6 is comparable to decaying all the way to 0, while leaving room for post-training. Hence, our final chosen annealing strategy is cosine LR decay to $3e$-6, which we use for our final two-phase experiments in §4.

[7] The `CMC-Blend3`-30% result at the bottom of Table 28 is also using 30% upsampling with LR cosine decay to $3e-6$.

Exp.	ARC-Easy	ARC-Challenge	RACE	PIQA	WinoGrande	HellaSwag	OpenBookQA	CommonsenseQA	Avg.
$\mathcal{P}_1$-Blend1	75.97	51.19	36.36	80.96	67.64	76.23	44.00	53.07	59.12
$\mathcal{P}_1$-Blend4	77.23	53.24	36.46	80.47	68.35	76.48	44.40	53.15	59.54
$\mathcal{P}_1$-Blend1-$\mathcal{P}_2$-Blend1	78.32	51.54	36.75	79.76	66.54	76.44	43.80	61.67	60.04
$\mathcal{P}_1$-Blend1-$\mathcal{P}_2$-Blend2	78.79	53.07	35.69	80.79	67.09	76.52	43.40	61.18	60.15
$\mathcal{P}_1$-Blend1-$\mathcal{P}_2$-Blend3	79.29	53.16	36.27	79.76	66.77	76.43	42.80	61.43	60.08
$\mathcal{P}_1$-Blend1-$\mathcal{P}_2$-Blend4	78.37	52.99	36.65	80.30	66.93	76.67	43.80	61.92	60.26
$\mathcal{P}_1$-Blend1-$\mathcal{P}_2$-Blend5	79.21	52.56	36.75	80.63	67.25	76.64	44.00	61.83	60.41
$\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend1	80.30	54.95	35.50	79.98	68.35	76.55	43.80	56.59	60.18
$\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend2	79.92	54.10	35.50	80.20	67.96	76.75	43.80	56.35	60.00
$\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend3	79.92	54.27	35.12	80.20	67.56	76.39	44.20	57.08	60.08
$\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend4	79.92	54.27	35.89	80.47	67.25	76.79	43.80	56.18	59.98
$\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend5	79.29	54.44	37.03	79.98	67.80	76.92	44.80	57.41	60.35

Table 18: Final reasoning evaluation results after $\mathcal{P}_2$ of training, broken down by individual benchmark. Corresponds to Table 7 in §4.

Exp.	MMLU					Code				
	STEM	Humanities	Social Sciences	Others	Avg.	HumanEval	HumanEval+	MBPP	MBPP+	Avg.
$\mathcal{P}_1$-Blend1	45.61	50.24	65.29	61.54	55.00	18.90	13.41	31.52	30.42	23.56
$\mathcal{P}_1$-Blend4	48.30	49.88	67.53	62.79	56.25	18.90	16.46	42.80	32.28	27.61
$\mathcal{P}_1$-Blend1-$\mathcal{P}_2$-Blend1	47.57	51.12	65.88	62.34	56.04	32.32	27.44	42.41	42.59	36.19
$\mathcal{P}_1$-Blend1-$\mathcal{P}_2$-Blend2	48.65	50.41	65.78	61.67	55.88	31.71	26.83	42.41	42.59	35.89
$\mathcal{P}_1$-Blend1-$\mathcal{P}_2$-Blend3	47.61	50.69	65.78	61.96	55.80	31.10	25.61	43.97	42.33	35.75
$\mathcal{P}_1$-Blend1-$\mathcal{P}_2$-Blend4	48.11	50.84	65.81	62.76	56.15	28.66	25.61	42.80	42.86	35.59
$\mathcal{P}_1$-Blend1-$\mathcal{P}_2$-Blend5	48.94	51.41	66.14	62.28	56.49	28.66	23.78	42.02	43.12	34.40
$\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend1	49.29	50.35	67.44	62.66	56.58	31.10	24.39	49.42	43.12	37.01
$\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend2	49.44	50.92	67.47	62.99	56.89	30.49	25.00	49.81	42.59	36.97
$\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend3	49.19	49.37	66.88	62.60	56.10	27.44	20.73	48.25	44.18	36.15
$\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend4	49.16	50.84	68.35	63.18	57.03	31.71	23.17	47.47	42.86	36.30
$\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend5	49.38	50.86	68.18	62.60	56.92	28.66	20.12	45.53	43.92	34.56

Table 19: Final MMLU and code evaluation results after $\mathcal{P}_2$ of training, broken down by individual category/benchmark. Corresponds to Table 7 in §4.

Exp.	ARC-Easy	ARC-Challenge	RACE	PIQA	WinoGrande	HellaSwag	OpenBookQA	CommonsenseQA	Avg.
BASE	78.75	53.84	35.69	80.30	68.51	76.30	45.40	51.68	59.69
Two-Phase	80.30	54.95	35.50	79.98	68.35	76.55	43.80	56.59	60.18

Table 20: Reasoning evaluation results of our two-phase training approach with $\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend1 vs. a randomized mixture of both blends across the entire 1T token training run, broken down by individual benchmark. Corresponds to Table 4 in §4.

Exp.	MMLU					Code				
	STEM	Humanities	Social Sciences	Others	Avg.	HumanEval	HumanEval+	MBPP	MBPP+	Avg.
BASE	50.40	49.67	66.49	63.08	56.49	28.66	25.00	44.36	44.18	35.55
Two-Phase	49.29	50.35	67.44	62.66	56.58	31.10	24.39	49.42	43.12	37.01

Table 21: MMLU and code evaluation results of our two-phase training approach with $\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend1 vs. a randomized mixture of both blends across the entire 1T token training run, broken down by individual category/benchmark. Corresponds to Table 4 in §4.

Exp.	Tok.	ARC-Easy	ARC-Challenge	RACE	PIQA	WinoGrande	HellaSwag	OpenBookQA	CommonsenseQA	Avg.
$\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend1	1T	80.30	54.95	35.50	79.98	68.35	76.55	43.80	56.59	60.18
$\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend1	1.7T	79.84	52.90	36.27	79.65	70.01	77.79	44.80	61.75	60.88
$\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend6	1.7T	80.09	55.29	37.03	80.79	70.09	78.53	46.00	60.85	61.63

Table 22: Reasoning evaluation results of scaling for 1.7T tokens vs. 1T tokens, with and without high-quality data epoch adjustment, broken down by individual benchmark. Corresponds to Table 8 in §4.

Exp.	Tok.	MMLU					Code				
		STEM	Humanities	Social Sciences	Others	Avg.	HumanEval	HumanEval+	MBPP	MBPP+	Avg.
$\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend1	1T	49.29	50.35	67.44	62.66	56.58	31.10	24.39	49.42	43.12	37.01
$\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend1	1.7T	51.22	52.41	68.74	65.47	58.61	31.10	25.61	48.25	45.50	37.62
$\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend6	1.7T	52.39	53.41	70.20	66.91	59.85	37.20	28.66	47.08	45.50	39.61

Table 23: MMLU and code evaluation results of scaling for 1.7T tokens vs. 1T tokens, with and without high-quality data epoch adjustment, broken down by individual category/benchmark. Corresponds to Table 8 in §4.

Model Size	ARC-Easy	ARC-Challenge	RACE	PIQA	WinoGrande	HellaSwag	OpenBookQA	CommonsenseQA	Avg.
8B	80.60	53.50	37.22	80.20	70.17	76.57	45.40	61.67	61.16
25B	82.74	57.59	37.13	81.07	72.38	78.62	47.20	68.55	63.29

Table 24: Reasoning evaluation results for 8B vs. 25B parameter models, using the same blend: $\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend1, broken down by individual benchmark. Note that we use a maximum sequence length of 8192 (instead of 4096) for both models here. Corresponds to Table 9 in §4.

Model Size	MMLU					Code				
	STEM	Humanities	Social Sciences	Others	Avg.	HumanEval	HumanEval+	MBPP	MBPP+	Avg.
8B	50.05	50.82	67.70	64.21	57.31	32.93	28.66	47.47	46.83	38.97
25B	58.07	59.30	77.71	72.48	65.97	37.20	33.54	58.37	53.17	45.57

Table 25: MMLU and code evaluation results for 8B vs. 25B parameter models, using the same blend: $\mathcal{P}_1$-Blend4-$\mathcal{P}_2$-Blend1, broken down by individual category/benchmark. Note that we use a maximum sequence length of 8192 (instead of 4096) for both models here. Corresponds to Table 9 in §4.

Category	Domain	CMC-B1	CMC-B2	CMC-B3
Web Crawl	-	30	15	30
High-Quality	Math	23	33	33
	Wiki	2	2	2
	Code	25	30	15
Medium-Quality	Books	9	9	9
	Papers	11	11	11
	$\texttt{CC}_{\texttt{dv}}$	0	0	0
Multilingual	-	0	0	0

Table 26: Finegrained CMC $\mathcal{P}_2$ Blends (in %), part 1.

Category	Domain	CMC-B3-F10ep	CMC-B3-F20ep	CMC-B3-GSM8K	Combo
Web Crawl	-	27.1	24.2	30	28.3
High-Quality	Math	33	33	31	33
	Wiki	2	2	2	2
	Code	15	15	15	15
Medium-Quality	Books	9	9	9	9
	Papers	11	11	11	11
	$\texttt{CC}_{\texttt{dv}}$	0	0	0	0
Multilingual	-	0	0	0	0
Task Data	FLAN	2.9	5.8	0	1
	GSM8K	0	0	2	0.7

Table 27: Finegrained CMC $\mathcal{P}_2$ Blends (in %), part 2.

Blend/Exp.	MMLU	Reason.	GSM8K	Code	Avg.
$\mathcal{P}_1$-only	56.10	60.64	16.60	16.22	44.48
CMC-B1	49.49	57.79	16.30	21.13	43.76
CMC-B2	55.92	60.52	22.97	21.80	46.45
CMC-B3	56.33	60.48	22.59	21.53	46.34
CMC-B3-F10ep	56.45	62.80	25.70	21.46	47.89
CMC-B3-F20ep	56.75	62.49	26.84	22.05	47.98
CMC-B3-GSM8K	56.27	60.65	35.56	21.52	47.37
CMC-B3-30%	56.30	59.61	32.15	23.51	47.10
Combo	56.22	62.51	45.19	25.58	**50.27**

Table 28: Results of finegrained $\mathcal{P}_2$ experiments. Code results here average across only HumanEval and MBPP, but not the + versions of both. Hence, they are not directly comparable with the paper results elsewhere.

Decay Strategy	Final LR	MMLU	Reason.	GSM8K	Code	Avg.
Linear	0	56.33	61.78	22.59	21.53	46.34
Linear	$3e-6$	56.16	61.63	23.35	20.68	46.08
Cosine	0	56.25	61.74	21.91	21.18	46.19
Cosine	$3e-6$	56.44	61.79	23.05	20.75	46.17

Table 29: Results of different learning rate annealing strategies for $\mathcal{P}_2$.