FRAME: Boosting LLMs with
A Four-Quadrant Multi-Stage Pretraining Strategy

Xuemiao Zhang[1,4*], Feiyu Duan[2,4*], Liangyu Xu[4*], Yongwei Zhou[4],
Sirui Wang[3,4†], Rongxiang Weng[4], Jingang Wang[4], Xunliang Cai[4]

[1] Peking University   [2] Beihang University   [3] Tsinghua University   [4] Meituan

zhangxuemiao@pku.edu.cn   duanfeiyu@buaa.edu.cn   ywzhouphd2018@gmail.com
{xuliangyu02, wangsirui, wangjingang02, caixunliang}@meituan.com

Abstract

Large language models (LLMs) have significantly advanced human language understanding and generation, with pretraining data quality and organization being crucial to their performance. Multi-stage pretraining is a promising approach, but existing methods often lack quantitative criteria for data partitioning and instead rely on intuitive heuristics. In this paper, we propose the novel **F**our-quad**RA**nt **M**ulti-stage pr**E**training strategy (FRAME), guided by the established principle of organizing the pretraining process into four stages to achieve significant loss reductions four times. This principle is grounded in two key findings: first, training on high Perplexity (PPL) data followed by low PPL data, and second, training on low PPL difference (PD) data followed by high PD data, both causing the loss to drop significantly twice and performance enhancements. By partitioning data into four quadrants and strategically organizing them, FRAME achieves a remarkable **16.8%** average improvement over random across MMLU and CMMLU for the 3B model, effectively boosting LLM performance.

1 Introduction

LLMs have significantly advanced human language understanding and generation (Touvron et al., 2023; Dubey et al., 2024; Islam and Moushi, 2024). The quality and organization of pretraining data are crucial as they directly impact LLM performance. Studies show that LLMs require different data at various pretraining stages (Yu et al., 2024), offering key insights for developing more efficient pretraining data organization strategies.

Recent studies partition the pretraining process into multiple stages, allowing models to learn from data with distinct characteristics at each stage, which enhances pretraining efficiency(Liu et al., 2021a; Yıldız et al., 2024; Anonymous, 2025). For

---

[*]Equal contribution.
[†]Corresponding author.

(a) Accuracy on MMLU.   (b) Accuracy on CMMLU.

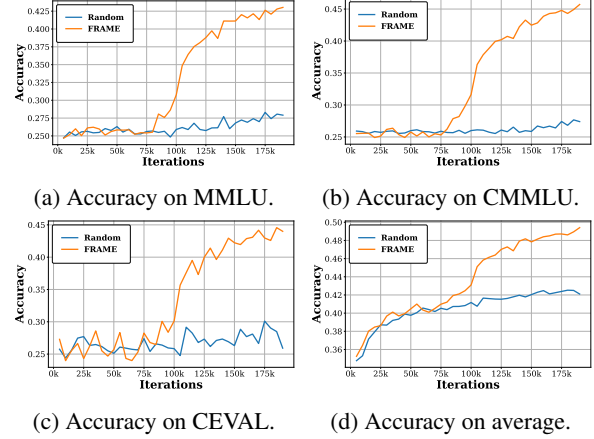(c) Accuracy on CEVAL.   (d) Accuracy on average.

Figure 1: Few-shot downstream performance on various benchmarks with respect to pretraining iterations, for 3B models trained on 1T tokens. The average performance is based on 8 downstream tasks. FRAME achieves 15.3% improvement over Random across MMLU, and 18.2% on CMMLU.

instance, MSP introduces specific tasks and data at each phase, allowing models to learn language structures and semantics from simple to complex progressively (Liu et al., 2021a). However, these methods don't provide quantitative criteria for partitioning data across stages, often relying on intuitive heuristics. The limitation underscores the need for more systematic approaches to optimize multi-stage pretraining.

In this paper, we propose a novel **F**our-quad**RA**nt **M**ulti-stage pr**E**training strategy (FRAME) to boost LLMs' performance. This strategy is systematically guided by the principle of organizing the pretraining process into four stages to achieve significant loss drops four times. The principle is derived from two key findings based on quantifiable data metrics.

Specifically, we intuitively use the metric Perplexity (PPL) to partition the data into two parts: high PPL and low PPL. Our first key finding reveals that training on high PPL data first, followed by low
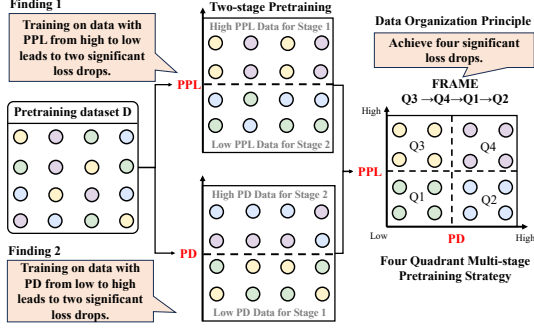
Figure 2: The overall framework of FRAME.

PPL data, leads to the loss dropping significantly twice and boosts model performance. Inspired by PDPC (Zhang et al., 2025), we also introduce the PPL difference (PD) between strong and weak models as another metric for data partitioning. Our second key finding shows that training on low PD data first, followed by high PD data, similarly causes the pretraining loss to drop significantly twice, thereby boosting model performance.

Based on these two major findings, we establish the principle of **organizing the pretraining process into four stages to achieve significant loss reductions four times**. Specifically, we partition the data into four quadrants based on PPL and PD: Quadrant 1 (Q_1) with low PD and low PPL, Quadrant 2 (Q_2) with high PD and low PPL, Quadrant 3 (Q_3) with low PD and high PPL, and Quadrant 4 (Q_4) with high PD and high PPL. Through further analysis and experiments, we determine that the optimal strategy is to reorganize the pretraining data in the $Q3 \rightarrow Q4 \rightarrow Q1 \rightarrow Q2$ sequence. To ensure smooth data transitions between stages, we also implement a smoothing process for gradual stage transitions. All data is processed offline to ensure the continuity of model training is not disrupted. Practical evidence shows that FRAME results in loss dropping significantly four times, enhances the models' emergent abilities, and boosts their performance. As shown in Figure 1, FRAME achieves a significant average improvement of 15.3% on MMLU and 18.2% on CMMLU over random selection, respectively.

In summary, our contributions are as follows: (1) We identify two key findings: first, training on high PPL data followed by low PPL data, and second, training on low PD data followed by high PD data, both causing two significant loss reductions and performance enhancements. (2) Guided by the principle of continuous loss reduction, we pro-

pose FRAME, a novel strategy that partitions data into four quadrants and organizes the pretraining process into four stages, resulting in loss dropping significantly four times and boosting model performance. (3) Experiments on 3B model, trained on 1T tokens, demonstrate a **16.8%** improvement over uniform sampling in average performance across MMLU and CMMLU.

2 Four Quadrant Multi-stage Pretraining Strategy

In this section, we first evaluate the effectiveness of PPL and PD as metrics for data organization in two-stage training (Sections 2.1 and 2.2). Based on these insights, we propose FRAME with both metrics(Section 2.3), as shown in Figure 2.

2.1 Two-stage pretraining guided by PPL

Research has explored pre-determining the sequence of pretraining data points based on their characteristics, which helps optimize models to globally optimal solutions (Pattnaik et al., 2024; Soviany et al., 2022). A critical aspect of this process is selecting appropriate metrics to organize data effectively, thereby minimizing training loss. Characteristics for text data include length, rare word frequency, and syntactic structure (Campos, 2021). Although these heuristic methods seem reasonable from a human cognitive perspective, they may not necessarily align with the specific requirements of the model. Thus, data characteristics should be determined using metrics that are perceptible to the model and align with the standards of the target tasks (Xu et al., 2020). In pretraining tasks, PPL closely aligns with the self-supervised learning objective (language modeling) and effectively evaluates model-data fit, making it an appropriate metric for organizing data.

Experimental Setting We extract 500B tokens from a bilingual dataset, with both English and Chinese corpora [1]. We train a 1.3B reference model (RM) on the subset using a random sequence and compute PPL of the subset using the RM. Based on the median PPL of the dataset, we partition the training data into two equal subsets: $A_{\text{PPL}}^{\text{low}}$ and $A_{\text{PPL}}^{\text{high}}$. The data within each subset is uniformly distributed. We conduct two-stage training on the 3B model in the sequences $A_{\text{PPL}}^{\text{low}} \rightarrow A_{\text{PPL}}^{\text{high}}$ and $A_{\text{PPL}}^{\text{high}} \rightarrow A_{\text{PPL}}^{\text{low}}$, and compare the results with

---

[1]For the details of the dataset, please refer to Section 3.1

Methods	MMLU	CMMLU
Random	24.8	25.6
$A_{PPL}^{low} \rightarrow A_{PPL}^{high}$	26.0	26.0
$A_{PPL}^{high} \rightarrow A_{PPL}^{low}$	**39.6**	**42.6**

Table 1: Accuracy on MMLU and CMMLU for two-stage pretraining based on PPL with 3B models.

those from the random training model. We evaluate model performance using MMLU(Hendrycks et al., 2020) and CMMLU(Li et al., 2023).
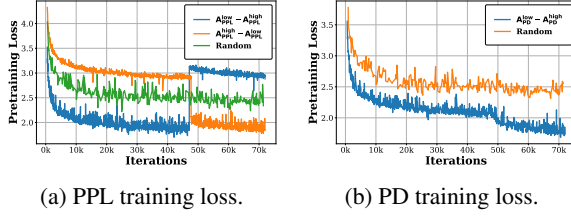

(a) PPL training loss.   (b) PD training loss.

Figure 3: Two-stage pretraining losses based on PPL and PD, respectively.

Results   Figure 3a illustrates the changes in training loss over time. It is observed that training on high PPL data followed by low PPL data results in significant loss reductions occurring twice, ultimately achieving a lower loss level. Conversely, the reverse setting maintains a higher loss, which remains above that of the Random setting. Table 1 shows the model's benchmark accuracy for the settings $A_{PPL}^{low} \rightarrow A_{PPL}^{high}$ and $A_{PPL}^{high} \rightarrow A_{PPL}^{low}$. Training on high PPL data followed by low PPL data yields significantly higher performance than random training. In contrast, in the $A_{PPL}^{low} \rightarrow A_{PPL}^{high}$ setting, the model shows only slight improvement. This leads to our **first key finding: training first on high PPL data followed by low PPL data can cause the loss to drop significantly twice, ultimately boosting model performance.**

2.2 Two-stage pertaining guided by PD

Subsequent analysis (detailed in Section 3.4) reveals that relying solely on PPL as a metric presents issues, as it is not stable across diverse data domains. Inspired by PDPC (Zhang et al., 2025), PD between strong and weak models can also reflect the difficulty of samples for the models. Consider two models, the weak model M_w and the strong model M_s, both trained on an identical dataset D. For any sample x, we calculate PD as follows:

$$PD(x) = \frac{PPL_{M_w}(x) - PPL_{M_s}(x)}{PPL_{M_w}(x)} \quad (1)$$

Methods	MMLU	CMMLU
Random	24.8	25.6
$A_{PD}^{low} \rightarrow A_{PD}^{high}$	**26.9**	**27.2**

Table 2: Accuracy on MMLU and CMMLU for two-stage pretraining based on PD with 3B models.

where $PPL_{M_w}(x)$ and $PPL_{M_s}(x)$ are the perplexity of the sample x calculated using M_w and M_s, respectively. A low PD indicates similar learning efficiency for both models, while a high PD suggests the sample is more challenging for the weak model.

If checkpoints from earlier and later training stages of the same model are viewed as weak and strong models (with the same parameters but improved performance due to more data in later stages), then data with low PD values pose similar difficulty for both early and late stages, while data with high PD values are more challenging for the model's early checkpoints. In light of the above analysis, PD emerges as a model-aware difficulty metric that is well-suited for organizing text data.

The distributions of PPL and PD on different domains are analyzed in Appendix B.5, PD exhibits a relatively consistent distribution across varied domains, following a normal distribution with an approximate mean value of 0.3. Compared to PPL, PD offers a better advantage in maintaining data diversity throughout each stage of training.

Experimental Setting   We train a 100M parameter RM and then calculate the PD for each sample using both the 100M and 1.3B RMs, referred to as PD(100M-1.3B). Using the median value of PD across all data, we divide the training dataset into two subsets with equal token counts: A_{PD}^{low} and A_{PD}^{high}. Based on PDPC's finding that a low-to-high PD ordering achieves better results, we conduct a two-stage training process with A_{PD}^{low} first, followed by A_{PD}^{high}, and compare the results to the random setting. We use the evaluation methods described in Section 2.1.

Results   Figure 3b shows the training loss changes under the $A_{PD}^{low} \rightarrow A_{PD}^{high}$ setup. The loss initially drops rapidly during the low PD phase, then stabilizes, and decreases further with high PD data, eventually falling below the Random model's loss. It suggests that the first phase sets a beneficial optimization path for the second, helping avoid local optima. Table 2 shows the model's accuracy

under the $A_{PD}^{low} \rightarrow A_{PD}^{high}$ setting. Notably, it exceeds the Random setting by 2.1% on MMLU and 1.6% on CMMLU, which validates PD as an effective metric. Further, this leads to our **second key finding: training first on low PD data followed by high PD data can cause the loss to drop significantly twice, ultimately boosting model performance.**

2.3 Four-Quadrant Guided Training Strategy

Inspired by the two key findings about PPL and PD, we establish the principle of **organizing pretraining data to achieve significant reductions in training loss**. Based on this principle, we introduce a novel **F**our quad**RA**nt **M**ulti-stage pr**E**training strategy (FRAME), which uses PPL and PD to partition data and reorganize the data sequence to make the training loss drop significantly. The core process of FRAME is shown in Algorithm 1.

Specifically, we train two RMs on the target training set D, with the strong model M_s having more parameters than the weak model M_w. Both models are trained on data from the same distribution and under identical settings. We compute PPL using M_s and PD using both M_s and M_w for each data point in D. We use the two metrics to partition $\mathcal{D}$ into four quadrants, as shown in Figure 4a. Our main goal is to ensure that the token numbers of the four quadrants are roughly equal. Therefore, we first determine the PPL threshold to divide the entire dataset into two parts with equal token numbers. Then, for both the high PPL and low PPL parts, we separately find each of their respective PD thresholds to further divide them into two sub-parts with equal token numbers, resulting in four quadrants. During the experiments, there are only slight differences in the PD thresholds for the two PPL subsets, and these dierences did not affect the final experimental results.

The data from four quadrants can be described as follows: Quadrant $\mathbf{Q_1}$ contains data that both M_s and M_w can fit well; Quadrant $\mathbf{Q_2}$ contains data that M_s learns well, but M_w struggles with. Data from this subset is challenging and requires a higher model capacity to understand them; Quadrant $\mathbf{Q_3}$ contains data poorly learned by both M_s and M_w; Quadrant $\mathbf{Q_4}$ contains data poorly learned by M_s and even worse by M_w.

Based on the two key findings about PPL and PD, we should follow these constraints:

$$\forall x \in S_i, \forall y \in S_{i+1}, \quad \text{PPL}(x) \geq \text{PPL}(y) \quad (2)$$
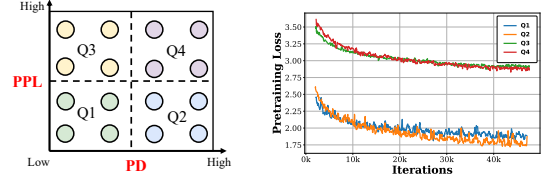
(a) Four Quadrants. (b) Losses of four quadrants.

Figure 4: Four-quadrant partitioning and pretraining losses of four quadrants.

$$\forall x \in S_i, \forall y \in S_{i+1}, \quad \text{PD}(x) \leq \text{PD}(y) \quad (3)$$

where S_i represents stage i of the training process. Equation (2) ensures that the PPL of data in stage i is not less than that in stage $i + 1$. Equation (3) ensures that the PD of data in stage i is not greater than that in stage $i + 1$.

To efficiently organize data from the four quadrants while adhering to constraints, we use a double-loop approach: first, dividing the training into two main phases based on a specific constraint, then further splitting each main phase into two sub-phases according to another constraint. The approach yields two distinct four-stage training strategies: the first strategy follows the sequence $Q_3 \rightarrow Q_4 \rightarrow Q_1 \rightarrow Q_2$, while the second adopts $Q_3 \rightarrow Q_1 \rightarrow Q_4 \rightarrow Q_2$. Both strategies break the constraints between the second and third stages. However, experiments in Sections 2.1 and 2.2 show that the loss change from low to high PPL is much greater than from high to low PD. Thus, we prioritize the constraints in Equation (2), making the first four-stage training strategy $\mathbf{Q_3 \rightarrow Q_4 \rightarrow Q_1 \rightarrow Q_2}$ the better choice.

Formulation of FRAME with Stage Transition
As illustrated in Figure 6b, direct stage transitions cause performance fluctuations. We aim to facilitate a smooth transition between stages.

We start by outlining the smoothing process f_{merge} for two-stage training: given the need to train on mixed data from two subsets D_1 and D_2, let i be the current training step and m the total number of training steps. The completion ratio is defined as $p = \frac{i}{m}$. The sampled batch B_i should satisfy the following condition:

$$B_i = \{x \mid x \sim D_1\}_{f(p) \cdot N} \cup \{x \mid x \sim D_2\}_{(1-f(p)) \cdot N} \quad (4)$$

where N represents the batch size, and $f(p)$ denotes the proportion of samples from D_1. Inspired by PDPC (Zhang et al., 2025), we employ the S-shape function as $f(p)$:

Algorithm 1 Ordered dataset construction via FRAME

1: **Input:** pretraining data D, batch size N, steepness a
2: **Output:** ordered dataset S_{frame}
3: **function** MERGEDATASETS(D_1, D_2, RandomSample)
4:     Total training steps $m \leftarrow \frac{|D_1|+|D_2|}{N}$
5:     **if** RandomSample **then**
6:         Randomly shuffle D_1 and D_2
7:     **end if**
8:     Initialize $l \leftarrow 0$, $r \leftarrow 0$, $Q_{\text{merged}} \leftarrow []$
9:     **for** i = 1 **to** m **do**
10:         Calculate completion ratio $p \leftarrow \frac{i}{m}$
11:         Calculate proportion $f(p) \leftarrow \frac{1}{1+\exp(a(p-0.5))}$
12:         $B_1 \leftarrow D_1[l : l + f(p)N]$
13:         $B_2 \leftarrow D_2[r : r + (1 - f(p))N]$
14:         Batch $B_i = B_1 \cup B_2$
15:         $l \leftarrow l + f(p)N$, $r \leftarrow r + (1 - f(p))N$
16:         Add all the samples in B_i to Q_{merged}
17:     **end for**
18:     **return** Q_{merged}
19: **end function**
20: Train RMs on i.i.d. subset of D
21: Calculate PPL and PD for all samples in D using RMs
22: Divide into 4 quadrants: Q_1, Q_2, Q_3, Q_4, based on PPL and PD thresholds.
23: $S_{34} \leftarrow$ MERGEDATASETS(Q_3, Q_4, True)
24: $S_{12} \leftarrow$ MERGEDATASETS(Q_1, Q_2, True)
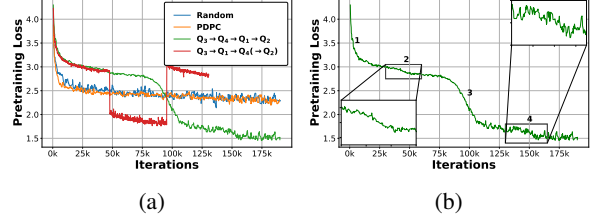25: $S_{\text{frame}} \leftarrow$ MERGEDATASETS(S_{34}, S_{12}, False)

Figure 5: Pretraining losses of main results. (a) Comparison of different methods. (b) Details of pretraining loss. The marked numbers indicate the process of the loss decreasing four times.

$$f(p) = \frac{1}{1 + \exp(a(p - 0.5))} \quad (5)$$

where a controls the steepness of the curve. Unlike PDPC, we merely utilize the S-shape function for smoothing during stage transitions. Therefore, we use a larger $a = 35$ instead of $a = 10$ as in PDPC, to achieve a steeper function curve, as illustrated in Appendix B.4.

In the four-stage training of FRAME, we initially obtain $S_{34} = f_{merge}(Q_3, Q_4)$ and $S_{12} = f_{merge}(Q_1, Q_2)$. During this phase, samples are randomly selected from the two data sources for each batch. Subsequently, we construct $S_{frame} = f_{merge}(S_{34}, S_{12})$. In this phase, samples must be drawn in the original sequence of S_{12} and S_{34} to maintain the order of $Q_3 \rightarrow Q_4 \rightarrow Q_1 \rightarrow Q_2$.

Notably, FRAME only organizes the given data without performing selection, which can serve as the final data preprocessing step before pretraining, without disrupting the continuity of training.

3 Experiments

3.1 Settings

Data Source For 3B models, our pretraining data is derived from various domains, including books (Gao et al., 2020), blogs (Baumgartner et al., 2020), patents (Sharma et al., 2019), Common Crawl (Penedo et al., 2024), and Wikipedia. It comprises

a total of 1T tokens, with 500B tokens each for Chinese and English, akin to the Matrix dataset (Zhang et al., 2024a). For 1.3B models, we randomly select 100B tokens from the SlimPajama dataset (Soboleva et al., 2023).

Pretraining Setting We train 100M and 1.3B models on i.i.d subsets of the collected dataset, comprising 500B tokens, to compute PPL and PD. And we test on 3B models for the main experiments, with a batch size of 640 and a context window length of 8192. The Adam optimizer is used for training within the Megatron framework. In addition, we validate our approach on a 1.3B model, employing the same configuration. More details of model structure and training could be found in Appendix B.1.

Baselines We compare FRAME with random data sequence and PDPC (Zhang et al., 2025). In addition we also validate on a 1.3B model, comparing it with other models based on PPL, PD, and QuRating metrics. More details of baselines could be found in Appendix B.1.

Evaluation For 1.3B models, we evaluate the models' performance on the following benchmarks: ARC-E (Clark et al., 2018), ARC-C (Clark et al., 2018), SciQ (Welbl et al., 2017), HellaSwag (Zellers et al., 2019), and PIQA (Bisk et al., 2020). For 3B models, we additionally use benchmarks including MMLU (Hendrycks et al., 2020), CMMLU (Li et al., 2023), BBH (Suzgun et al., 2022), and CEVAL (Huang et al., 2023), covering more complicated tasks. We apply in-context learning and select examples based on task characteristics. Standard accuracy serves as the final metric for all tasks.

3.2 Main Results

Table 4 shows our experimental results. FRAME significantly outperforms the Random baseline,

Method	Metric	Order	ARC-E	ARC-C	SciQ	HellaSw.	PIQA	AVG.
-	-	Random	56.5	23.6	85.8	34.2	67.3	53.5
Sequential	PD	High2Low	54.7 ↓1.8	21.8 ↓1.8	87.1 ↑1.3	33.7 ↓0.5	67.8 ↑0.5	53.0 ↓0.5
	PD	Low2High	56.1 ↓0.4	21.3 ↓2.3	86.2 ↑0.4	34.4 ↑0.2	67.6 ↓0.2	53.1 ↓0.4
	PPL	High2Low	45.5 ↓11.0	20.6 ↓3.0	71.2 ↓14.6	30.3 ↓3.9	63.7 ↓3.6	46.3 ↓7.2
	PPL	Low2High	47.8 ↓8.7	17.9 ↓5.7	72.7 ↓13.1	29.1 ↓5.1	62.4 ↓4.9	46.0 ↓7.5
	Qu.Edu	High2Low	57.2 ↑0.7	26.4 ↑2.8	85.4 ↓0.4	33.0 ↓1.2	66.2 ↓1.1	53.6 ↑0.1
	Qu.Edu	Low2High	56.8 ↑0.3	26.0 ↑2.4	84.1 ↓1.7	33.5 ↓0.7	67.9 ↑0.6	53.7 ↑0.2
Preference CL	PPL	S.R.	56.1 ↓0.4	24.1 ↑0.5	87.8 ↑2.0	33.9 ↓0.3	67.4 ↑0.1	53.9 ↑0.4
	PPL	S.	56.1 ↓0.4	22.6 ↓1.0	85.5 ↓0.3	34.2 0.0	67.5 ↑0.2	53.2 ↓0.3
	Qu.Edu	S.R	56.7 ↑0.2	24.9 ↑1.3	86.2 ↑0.4	33.6 ↓0.6	66.9 ↓0.4	53.7 ↑0.2
	Qu.Edu	S.	55.5 ↓1.0	24.8 ↑1.2	87.8 ↑2.0	34.0 ↑0.2	67.4 ↑0.1	53.9 ↑0.4
	PD	S.R.	56.7 ↑0.2	24.9 ↑1.3	86.2 ↑0.4	33.6 ↓0.6	67.4 ↑0.1	53.8 ↑0.3
PDPC	PD	S.	57.3 ↑0.8	26.6 ↑3.0	87.9 ↑2.1	33.7 ↓0.5	68.0 ↑0.7	54.7 ↑1.2
FRAME	PPL&PD	-	**62.9** ↑6.4	**26.5** ↑2.9	**90.5** ↑4.7	**38.4** ↑4.2	**69.6** ↑2.3	**57.6** ↑4.1

Table 3: Downstream tasks results for different settings on **1.3B** models. We report accuracy for each task, and the best performances are marked in bold. Abbreviations: AVG. = Average, S.=S-shape Function, S.R.=S-shape Reverse Function.

Method	MMLU	CMMLU	CEVAL	BBH	ARC-E	ARC-C	HellaSw.	PIQA	AVG.
Random	27.7	27.5	27.2	27.9	68.6	33.7	49.4	76.0	42.3
PDPC	35.8 ↑8.1	35.6 ↑8.1	36.1 ↑8.9	25.7 ↓2.2	69.7 ↑1.1	35.8 ↑2.1	49.9 ↑0.5	76.3 ↑0.3	45.6 ↑3.3
$Q_3 \rightarrow Q_1 \rightarrow Q_4 \dashrightarrow Q_2$	25.8 ↓1.9	26.4 ↓1.1	25.5 ↓1.6	28.9 ↑1.0	68.3 ↓0.3	35.2 ↑1.5	48.3 ↓1.1	75.4 ↓0.6	41.7 ↓0.6
FRAME	**43.0** ↑15.3	**45.7** ↑18.2	**44.0** ↑16.8	27.9 0.0	**71.0** ↑2.4	**36.5** ↑2.8	**50.2** ↑0.8	**76.9** ↑0.9	**49.4** ↑6.1

Table 4: Results of downstream tasks for different methods using **3B** models on **1T** tokens. "--→" indicates that the model's accuracy has significantly decreased before reaching this stage, so we stopped at the third stage.

with a 6.1% improvement in average performance. For specific tasks, it achieves notable gains of 15.3% on MMLU and 18.2% on CMMLU. For the 1.3B model, as illustrated in Table 3, FRAME also surpasses all the baselines, with 4.1% improvement in average, and a maximum of 6.4% gain in a single task. These findings validate the effectiveness of FRAME. Moreover, in our experiments with the 1.3B and 3B models, distinct pretraining datasets are utilized for each model. Notably, FRAME consistently yield substantial performance gains across all evaluated benchmarks. These results highlight the robustness and generalizability of our method, demonstrating its effectiveness across varying data sources and model architectures.

Figure 1 shows the model's benchmark performance over training steps. The model remains stable initially, with a significant performance increase after 90K steps. Figure 5 illustrates the training loss, showing four declines that align with benchmark performance improvements. This con-

firms the strong link between training loss and benchmark performance: when training stages are equally divided, lower training loss results in better performance. Additionally, FRAME effectively enhances the model's emergent capabilities, allowing it to acquire foundational skills quickly and better understand data patterns in later stages, improving overall performance. Through the loss smoothness analysis in Appendix B.7, we found that the loss curve of FRAME has the lowest high-frequency energy proportion of only 0.02%, significantly lower than Random and PDPC. This indicates that FRAME can make the model converge more stably and reduce the impact of gradient fluctuations during training.

We find that the $Q_3 \rightarrow Q_4 \rightarrow Q_1 \rightarrow Q_2$ strategy significantly outperforms the $Q_3 \rightarrow Q_1 \rightarrow Q_4 \rightarrow Q_2$ strategy, with the latter showing a performance decline in the third stage. This supports our argument that prioritizing the PPL dimension over the PD dimension is beneficial.
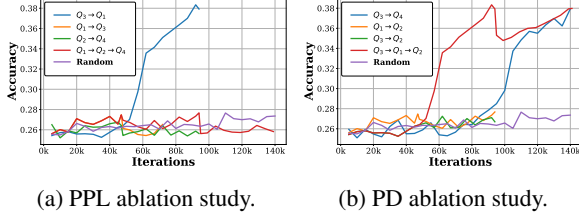
(a) PPL ablation study.　　(b) PD ablation study.

Figure 6: Ablation studies of different combinations.

3.3 Ablation Study

We investigate the differential effects of data from distinct quadrants and to characterize their intrinsic properties. Figure 4b presents the training loss trajectories across different quadrants. The results demonstrate a significant disparity in loss magnitude, with high PPL data (Q_3 and Q_4) exhibiting substantially greater loss values compared to their low PPL counterparts (Q_1 and Q_2). Furthermore, under equivalent PPL conditions, data with higher PD (Q_2 and Q_4) consistently enabled the model to converge to lower final loss values than data with lower PD (Q_1 and Q_3). Notably, our analysis reveals that PPL exerts a more substantial influence on training loss than PD.

Furthermore, we could observe a slower convergence rate when training on high PD data. This phenomenon can be attributed to the difficulty of high PD samples, which are particularly challenging for the model in the early stage of training when its capacity is still limited. However, as training progresses, the model's capabilities gradually improve. Since the weak and strong models fit low PD data similarly, even as the model enhances, the training loss will not further decrease, leading to faster convergence. On the other hand, high PD data, with the improvement of model abilities, can further guide the model to learn new features and continue reducing training loss.

From the perspective of four quadrants, do the conclusions of Sections 2.1 and 2.2 still hold? We test various quadrant combinations and find that:

Training first on high PPL data followed by low PPL data proves to be a superior strategy   Figure 6a illustrates the outcomes for each combination. We see a significant performance boost when transitioning from larger PPL data to smaller ones $Q_3 \rightarrow Q_1$. However, moving in the opposite direction $Q_1 \rightarrow Q_3$ and $Q_2 \rightarrow Q_4$ causes a certain degree of accuracy drop, regardless of whether PD is large or small. Over a longer training period, moving from $Q_1 \rightarrow Q_2$ to Q_4 also results in a
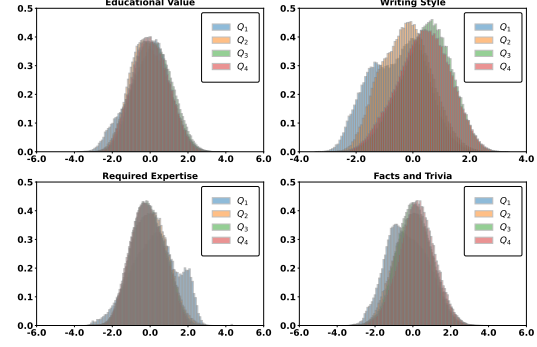

Figure 7: Human intuitive quality metric distribution.

performance decline.

Starting with low PD data followed by high PD data is more effective   As shown in Figure 6b, when PD increases ($Q_3 \rightarrow Q_4$, $Q_1 \rightarrow Q_2$, and $Q_3 \rightarrow Q_2$), the model performs better on the benchmark. Furthermore, in a longer training phase, transitioning from $Q_3 \rightarrow Q_1 \rightarrow Q_2$ can further enhance the model's performance.

In addition, the $Q_3 \rightarrow Q_1 \rightarrow Q_2$ setup, where f_{merge} is not applied, shows a performance drop between the second and third stages. This suggests that f_{merge} is necessary. More results can be found in Appendix B.8.

3.4 Analysis

Data in Quadrants Without Inherent Human Intuitive Cognition Favorability   We investigate the characteristics of data within the four quadrants by extracting 1,000 samples from each and analyzing their quality using four raters from QuRating. As shown in Figure 7, the quality distributions are similar across all dimensions for samples from each quadrant, which suggests that model-perceived measures like PPL and PD don't significantly correlate with human cognition-based measures, such as knowledge, quality, and diversity. FRAME maintains consistent data quality throughout the training process, preventing the model from focusing solely on low-quality data at any stage.

PPL and PD Distribution   We extract 70K samples from 7 domains within our training dataset, covering areas such as Arxiv, Law, Code, and Math. Using the 1.3B reference model, we calculate the PPL of all samples and present the PPL distributions in Figure 12. Significant differences are observed across various domains, with mean values ranging from 5 to 22 and varying degrees of variance. Data from domains such as Wikipedia typi-
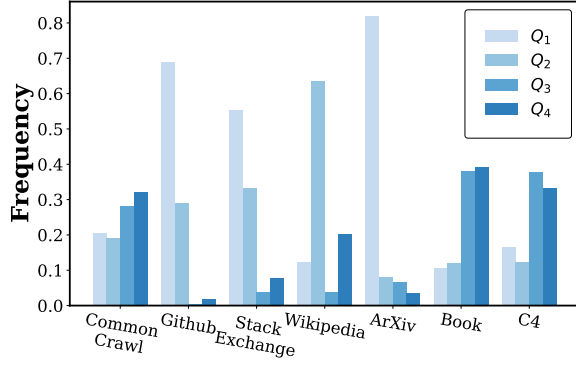
Figure 8: Data distribution across different sources.

Figure 9: Semantic analysis of different quadrants.

Figure 10: Semantic distribution of different quadrants.

cally exhibit lower PPL, whereas data from Reddit show higher PPL. This implies that sorting data by PPL could cause obvious shifts in domain representation between initial and later training stages, potentially exposing the model to overly homogeneous data during each training stage. Sachdeva et al. (2024) highlight that data diversity enhances model performance, and an uneven distribution may lead to a decline in performance.

We calculate the PD of the sampled data from the 100M and 1.3B reference models and visualize its distribution across various sources, as shown in Figure 13. We find that the PD distribution is quite similar across different sources. This similarity implies that using PD as a metric for data partitioning allows each training phase to include data from various sources, ensuring data diversity throughout the training process and thereby maintaining pretraining efficiency.

Data Distribution of the Four Quadrants We extract 1 million samples from each of the four quadrant distributions and visualize them, as shown in Figure 8. We observe that each quadrant contains data from a wide range of sources, indicating that FRAME effectively ensures data diversity throughout the training process. Notably, there are distinct differences among the quadrants. For instance, Q_2 contains a large number of domain-specific datasets such as Wikipedia, and StackExchange. This aligns with common cognitive understanding, as these datasets may feature more complex sentence structures and terminology, which large models can learn effectively while small models may struggle to master, thus suitable for learning in the later stages of training. This also explains the excellent performance in knowledge-intensive tasks, as the texts in Wikipedia contain a wealth of factual knowledge. Additionally, the data distribution in
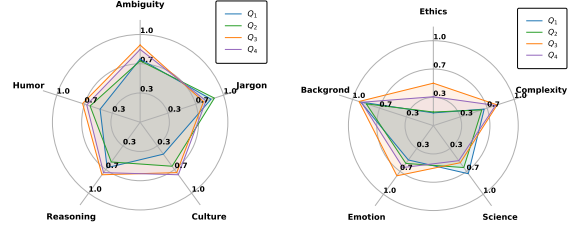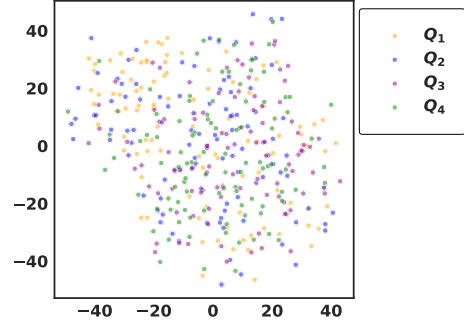
Q_3 is relatively balanced, which is advantageous for the initial training phase. Presenting data from Q_3 at the beginning of the training process allows the model to learn from sufficiently diverse data, thereby establishing a strong foundation for subsequent learning stages.

Semantic Properties Analysis To assess distinct semantic properties within each quadrant's data, we randomly select 1K samples from each quadrant and devise 10 language-text related traits for GPT-4o evaluation. For simplicity and precision, all traits are formulated as yes-or-no questions, and we calculate the percentage of samples meeting each trait's criteria. Details of all traits are listed in Appendix C. As shown in Figure 9, the data across all four quadrants exhibit similar semantic properties. In addition, we use T5 (Raffel et al., 2023) to obtain dense vectors of the samples and perform dimensionality reduction using t-SNE, as shown in Figure 10. We could observe that the data in the four quadrants do not have obvious distinctions at the semantic level. This indicates no obvious difference in the semantic distribution of data between the initial and later training stages, ensuring a diverse range of data is consistently encountered throughout the training process, and avoiding the collapse of the model into a certain preference, thereby damaging generalization performance.

4 Related Works

Multi-stage pretraining has emerged as a pivotal strategy in the development of LLMs, enabling models to better capture and utilize diverse data characteristics by dividing the training process into distinct phases(Pavlova, 2025; Zhao et al., 2024; Liu et al., 2021b; Tan et al., 2022). Liu et al. (2021b) propose a multi-stage pretraining method that leverages various granularities of information, significantly boosting model performance. Tan et al. (2022) explore the use of multi-stage prompting to improve translation tasks, demonstrating its effectiveness in enhancing downstream applications.

In LLM pretraining, data preprocessing is key to ensuring dataset quality(Duan et al., 2025). Traditional methods use expert rules to filter low-quality data (Raffel et al., 2020; Rae et al., 2021; Laurençon et al., 2022; Computer, 2023; Penedo et al., 2024) and remove duplicates (Lee et al., 2022; Sorscher et al., 2022; Abbas et al., 2023; Soboleva et al., 2023; Tirumala et al., 2024). These approaches enhance quality but may lack semantic depth. To improve semantic selection, strategies involve using targeted sources or proxy models (Wenzek et al., 2020; Xie et al., 2023; Marion et al., 2023; Thakkar et al., 2023; Engstrom et al., 2024; Yu et al., 2024). Classifiers automate selection, like the logistic regression model used by Du et al. (2022), with others employing complex scoring (Zhang et al., 2024b; Sachdeva et al., 2024). QuRating (Wettig et al., 2024) uses multiple raters for nuanced evaluation.

Current methods focus on selection but overlook aligning data characteristics with learning stages, missing opportunities in data organization and sequencing to boost pretraining effectiveness.

5 Conclusion

In this study, we propose the **F**our-quad**RA**nt **M**ulti-stage pr**E**training strategy (FRAME), a novel approach designed to boost the performance of LLMs by systematically organizing the pretraining process into four stages. Guided by the principle of achieving significant loss reductions four times, FRAME employs a strategic partitioning of pretraining data into four quadrants based on PPL and PD. The experimental results, showing a 16.8% average improvement across MMLU and CMMLU over random sampling, underscore the effectiveness of FRAME in optimizing pretraining

data organization.

6 Limitations and Future Works

In this study, we simply attempt to split the data into two parts based on the size of PPL and PD. Future research can be more detailed, such as subdividing a dimension into three parts or more, to explore more training stages. In addition, we plan to expand the scope of work to verify the applicability of the two-stage pretraining method in more language model architectures, such as Mamba(Gu and Dao, 2023), and Mixture of Experts(Wang et al., 2024). Although our evaluation criteria are already quite comprehensive, it is still possible to extend to a wider range of evaluations, including more detailed domain-specific or interactive tasks, such as the evaluation of analogy reasoning ability(Hu et al., 2023). Despite these potential limitations, we firmly believe that our research provides valuable insights and practical contributions to the academic community.

References

Amro Abbas, Kushal Tirumala, Dániel Simig, Surya Ganguli, and Ari S Morcos. 2023. Semdedup: Data-efficient learning at web-scale through semantic deduplication. *arXiv preprint arXiv:2303.09540*.

Anonymous. 2025. Multi-agent collaborative data selection for efficient language model pretraining.

Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. 2020. The pushshift reddit dataset. In *Proceedings of the international AAAI conference on web and social media*, volume 14, pages 830–839.

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.

Daniel Campos. 2021. Curriculum learning for language modeling. *arXiv preprint arXiv:2108.02170*.

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*.

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

Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. 2022.

Glam: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning*, pages 5547–5569. PMLR.

Feiyu Duan, Xuemiao Zhang, Sirui Wang, Haoran Que, Yuqi Liu, Wenge Rong, and Xunliang Cai. 2025. Enhancing llms via high-knowledge data selection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 23832–23840.

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

Logan Engstrom, Axel Feldmann, and Aleksander Madry. 2024. Dsdm: Model-aware dataset selection with datamodels. *arXiv preprint arXiv:2401.12926*.

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

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. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Xiaoyang Hu, Shane Storks, Richard L Lewis, and Joyce Chai. 2023. In-context analogical reasoning with pre-trained language models. *arXiv preprint arXiv:2305.17626*.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, Yao Fu, Maosong Sun, and Junxian He. 2023. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *Preprint*, arXiv:2305.08322.

Raisa Islam and Owana Marzia Moushi. 2024. Gpt-4o: The cutting-edge advancement in multimodal llm. *Authorea Preprints*.

Hugo Laurençon, Lucile Saulnier, Thomas Wang, Christopher Akiki, Albert Villanova del Moral, Teven Le Scao, Leandro Von Werra, Chenghao Mou, Eduardo González Ponferrada, Huu Nguyen, et al. 2022. The bigscience roots corpus: A 1.6 tb composite multilingual dataset. *Advances in Neural Information Processing Systems*, 35:31809–31826.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8424–8445.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2023. Cmmlu: Measuring massive multitask language understanding in chinese. *arXiv preprint arXiv:2306.09212*.

Tongtong Liu, Fangxiang Feng, and Xiaojie Wang. 2021a. Multi-stage pre-training over simplified multimodal pre-training models. *Preprint*, arXiv:2107.14596.

Tongtong Liu, Fangxiang Feng, and Xiaojie Wang. 2021b. Multi-stage pre-training over simplified multimodal pre-training models. *CoRR*, abs/2107.14596.

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*.

Pulkit Pattnaik, Rishabh Maheshwary, Kelechi Ogueji, Vikas Yadav, and Sathwik Tejaswi Madhusudhan. 2024. Curry-dpo: Enhancing alignment using curriculum learning & ranked preferences. *arXiv preprint arXiv:2403.07230*.

Vera Pavlova. 2025. Multi-stage training of bilingual islamic llm for neural passage retrieval. *arXiv preprint arXiv:2501.10175*.

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

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer. *Preprint*, arXiv:1910.10683.

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*.

Eva Sharma, Chen Li, and Lu Wang. 2019. Bigpatent: A large-scale dataset for abstractive and coherent summarization. *arXiv preprint arXiv:1906.03741*.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*.

Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. 2023. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama.

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.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *Preprint*, arXiv:2210.09261.

Zhixing Tan, Xiangwen Zhang, Shuo Wang, and Yang Liu. 2022. MSP: Multi-stage prompting for making pre-trained language models better translators. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6131–6142, Dublin, Ireland. Association for Computational Linguistics.

Megh Thakkar, Tolga Bolukbasi, Sriram Ganapathy, Shikhar Vashishth, Sarath Chandar, and Partha Talukdar. 2023. Self-influence guided data reweighting for language model pre-training. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2033–2045.

Kushal Tirumala, Daniel Simig, Armen Aghajanyan, and Ari Morcos. 2024. D4: Improving llm pretraining via document de-duplication and diversification. *Advances in Neural Information Processing Systems*, 36.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Siqi Wang, Zhengyu Chen, Bei Li, Keqing He, Min Zhang, and Jingang Wang. 2024. Scaling laws across model architectures: A comparative analysis of dense and moe models in large language models. *arXiv preprint arXiv:2410.05661*.

Johannes Welbl, Nelson F Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. *arXiv preprint arXiv:1707.06209*.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Édouard Grave. 2020. Ccnet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4003–4012.

Alexander Wettig, Aatmik Gupta, Saumya Malik, and Danqi Chen. 2024. Qurating: Selecting high-quality data for training language models. *arXiv preprint arXiv:2402.09739*.

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

Benfeng Xu, Licheng Zhang, Zhendong Mao, Quan Wang, Hongtao Xie, and Yongdong Zhang. 2020. Curriculum learning for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6095–6104.

Çağatay Yıldız, Nishaanth Kanna Ravichandran, Prishruit Punia, Matthias Bethge, and Beyza Ermis. 2024. Investigating continual pretraining in large language models: Insights and implications. *arXiv preprint arXiv:2402.17400*.

Zichun Yu, Spandan Das, and Chenyan Xiong. 2024. Mates: Model-aware data selection for efficient pretraining with data influence models. *arXiv preprint arXiv:2406.06046*.

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*.

Ge Zhang, Scott Qu, Jiaheng Liu, Chenchen Zhang, Chenghua Lin, Chou Leuang Yu, Danny Pan, Esther Cheng, Jie Liu, Qunshu Lin, Raven Yuan, Tuney Zheng, Wei Pang, Xinrun Du, Yiming Liang, Yinghao Ma, Yizhi Li, Ziyang Ma, Bill Lin, Emmanouil Benetos, Huan Yang, Junting Zhou, Kaijing Ma, Minghao Liu, Morry Niu, Noah Wang, Quehry Que, Ruibo Liu, Sine Liu, Shawn Guo, Soren Gao, Wangchunshu Zhou, Xinyue Zhang, Yizhi Zhou, Yubo Wang, Yuelin Bai, Yuhan Zhang, Yuxiang Zhang, Zenith Wang, Zhenzhu Yang, Zijian Zhao, Jiajun Zhang, Wanli Ouyang, Wenhao Huang, and Wenhu Chen. 2024a. Map-neo: Highly capable and transparent bilingual large language model series. *arXiv preprint arXiv: 2405.19327*.

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

Yifan Zhang, Yifan Luo, Yang Yuan, and Andrew C Yao. 2024b. Autonomous data selection with language models for mathematical texts. In *ICLR 2024 Workshop on Navigating and Addressing Data Problems for Foundation Models*.

Hang Zhao, Yifei Xin, Zhesong Yu, Bilei Zhu, Lu Lu, and Zejun Ma. 2024. Slit: Boosting audio-text pretraining via multi-stage learning and instruction tuning. *arXiv preprint arXiv:2402.07485*.

A Ethical Considerations

We utilized publicly available training corpora from the internet to train our models, which inevitably included biased or harmful content, raising concerns about the safety of the model-generated content. To mitigate this issue, we prioritized the selection of high-quality datasets and implemented rigorous data-cleansing processes to remove harmful elements. Additionally, considering the computationally intensive nature of LLM training and its potential environmental impact, this paper explores a multi-stage training approach, aiming to enhance resource efficiency and reduce environmental pollution during the training process.

B Experimental Details

B.1 Experimental settings

Model configuration and training We train reference models with 100M and 1.3B parameters on 500B tokens randomly sampled from the collected dataset, utilizing the Llama architecture (Touvron et al., 2023). We use the 1.3B model to compute PPL and both the 100M and 1.3B models to calculate PD. For the main experiments, we train a 3B model. We set the batch size to 640 and the context window length to 8192. The initial learning rate is 2×10^{-4}, with a warm-up phase of 375M tokens. We apply cosine learning rate scheduling with a weight decay of 0.1. We use the Adam optimizer to train the model within the Megatron framework (Shoeybi et al., 2019). Furthermore, we validate

Hyperparameter	Value
Precision	bfloat16
Layers	30
Hidden dimension	1920
Attention heads	15
Vocab size	131072
Sequence length	8192
Activation	SiLU
Position embedding	RoPE

Table 5: Model structure of 1.3B model

Hyperparameter	Value
Precision	bfloat16
Layers	32
Hidden dimension	2560
Attention heads	32
Vocab size	131072
Sequence length	8192
Activation	SiLU
Position embedding	RoPE

Table 6: Model structure of 3B model

our approach on a 1.3B model, using the same settings as the 3B model. We use Ascend 910B NPU for training. For the 3B model, we use 512 NPUs for training, each model takes over 180 hours; For the 1.3B models, we use 96 NPUs, each model takes 6 hours.

In Table 5 and 6, we present the model configuration of the 1.3B and 3B models.

Baselines In the 3B model setting, we compare FRAME with several baselines:

- **Random**: Data is organized in a completely random order, meaning there is no specific sequence or strategy for data input

- **PDPC** (Zhang et al., 2025): Utilizes PD as a model-aware curriculum learning indicator, organizing the data sequence through a preference function that aligns with the model's inherent preferences.

- $Q_3 \rightarrow Q_1 \rightarrow Q_4 \rightarrow Q_2$: Adopts the same smoothing process as FRAME but prioritizes satisfying PD-related constraints to organize the data order.

In the 1.3B model setting, we also compare FRAME with other baselines:

Methods	Steps	MMLU	CMMLU
Random		25.4	25.9
$A_{PPL}^{low} \rightarrow A_{PPL}^{high}$	30K	**26.7**	**26.2**
$A_{PPL}^{high} \rightarrow A_{PPL}^{low}$		25.3	25.5
Random		25.9	25.8
$A_{PPL}^{low} \rightarrow A_{PPL}^{high}$	60K	25.5	25.5
$A_{PPL}^{high} \rightarrow A_{PPL}^{low}$		**33.4**	**35.4**
Random		24.8	25.6
$A_{PPL}^{low} \rightarrow A_{PPL}^{high}$	95K	26.0	26.0
$A_{PPL}^{high} \rightarrow A_{PPL}^{low}$		**39.6**	**42.6**

Table 7: Accuracy on MMLU and CMMLU for two-stage pretraining across different steps based on PPL with 3B models.

Methods	Steps	MMLU	CMMLU
Random	30K	25.4	25.9
$A_{\text{PD}}^{\text{low}} \to A_{\text{PD}}^{\text{high}}$		**26.7**	**26.2**
Random	60K	25.9	**25.8**
$A_{\text{PD}}^{\text{low}} \to A_{\text{PD}}^{\text{high}}$		**26.2**	25.1
Random	95K	24.8	25.6
$A_{\text{PD}}^{\text{low}} \to A_{\text{PD}}^{\text{high}}$		**26.9**	**27.2**

Table 8: Accuracy on MMLU and CMMLU for two-stage pretraining based on PD with 3B models.

- **Sequential**: Organizes data by directly sorting according to PPL, PD, and QuRating (Wettig et al., 2024), in either ascending or descending order.

- **Preference CL**: Utilizes the same preference function as PDPC but replaces the indicators with PPL or QuRating. We adopt both S-shape and S-shape reverse function.

B.2 Benchmark Accuracy of Two-stage Pretraining Based on PPL

Table 7 shows the model's benchmark accuracy at various steps for the settings $A_{\text{PPL}}^{\text{low}} \to A_{\text{PPL}}^{\text{high}}$ and $A_{\text{PPL}}^{\text{high}} \to A_{\text{PPL}}^{\text{low}}$. Training on high PPL data followed by low PPL data initially yields lower performance than random training at 30K steps, but accuracy improves significantly in the later phases at 60K and 95K steps. In contrast, in the $A_{\text{PPL}}^{\text{low}} \to A_{\text{PPL}}^{\text{high}}$ setting, the model shows slight improvement initially but eventually aligns with the random sequence results. This leads to our first key finding: training first on high PPL data followed by low PPL data can cause the loss to drop significantly twice, ultimately boosting model performance.
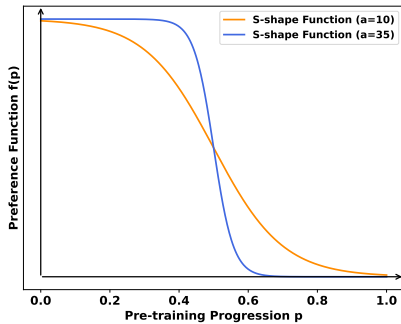
Figure 11: S-shape functions with a=10 and a=35.

B.3 Benchmark Accuracy of Two-stage Pretraining Based on PD

Table 8 shows the model's accuracy as training steps increase under the $A_{\text{PD}}^{\text{low}} \to A_{\text{PD}}^{\text{high}}$ setting. The model consistently outperforms the Random model at most steps. Notably, at 95K steps, it exceeds the Random setting by 2.1% on MMLU and 1.6% on CMMLU, which validates PD as an effective metric. This leads to our second key finding: training first on low PD data followed by high PD data can cause the loss to drop significantly twice, ultimately boosting model performance.

B.4 S-shape Functions with a=10 and a=35

By applying the S-shape function, we gradually decrease the proportion of Q_3 data while increasing the proportion of Q_4 data, resulting in the mixed data S_{34}. Figure 11 illustrates the two forms of Equation 5 with $a = 10$ and $a = 35$. In this paper, we adopt the steeper form with $a = 35$ for smoothing transitions between different stages. In the early stages of model training, the proportion of data from the first stage is close to 1. As training progresses, this proportion approaches 0. During the mid-training phase, there is a gradual transition between the data from the first and second stages.

B.5 Data Distribution

Figure 12 and 13 show the PPL and PD distribution of samples in our training set. The PPL of data from different domains shows significant variations, with mean values ranging from 5 to 22 and varying degrees of variance. Sorting data by PPL may lead to imbalanced domain representation between early and late training stages, exposing the model to overly homogeneous data at each stage and potentially degrading performance. In contrast, the distribution of PD is quite similar across different data sources, ensuring that each training phase includes diverse data, thereby maintaining pretraining efficiency and enhancing model performance.

Figure 14 illustrates the domain distribution of data across the four quadrants, arranged in descending order of proportion. Quadrants 1 and 2 predominantly contain data related to code (sourced from Github and Stack Exchange) and knowledge (sourced from Arxiv and Wikipedia). In contrast, the data in Quadrants 3 and 4 primarily originate from books and the Common Crawl.
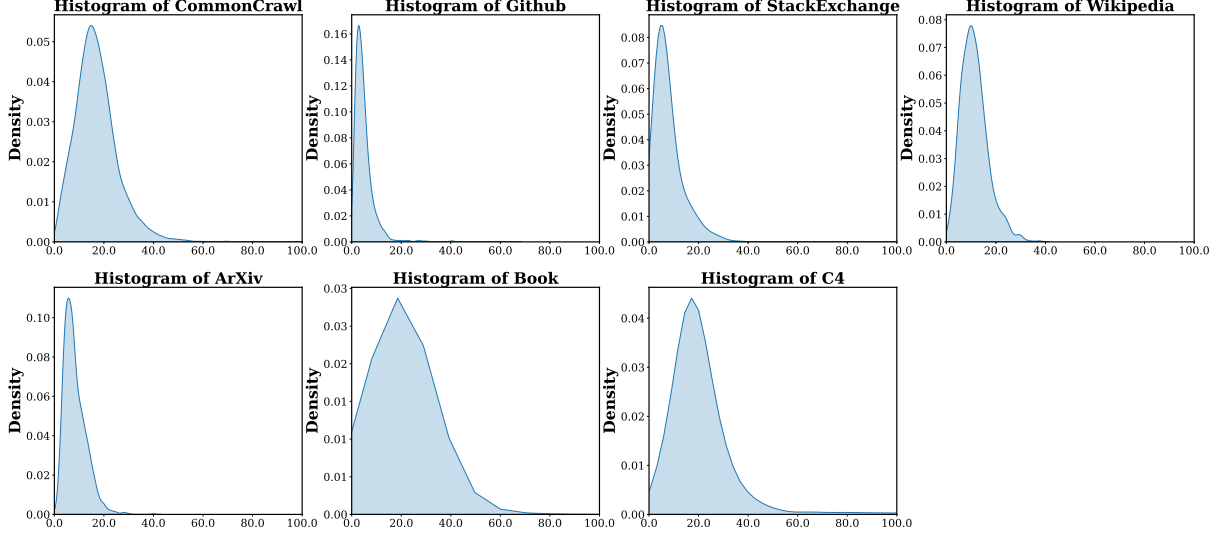
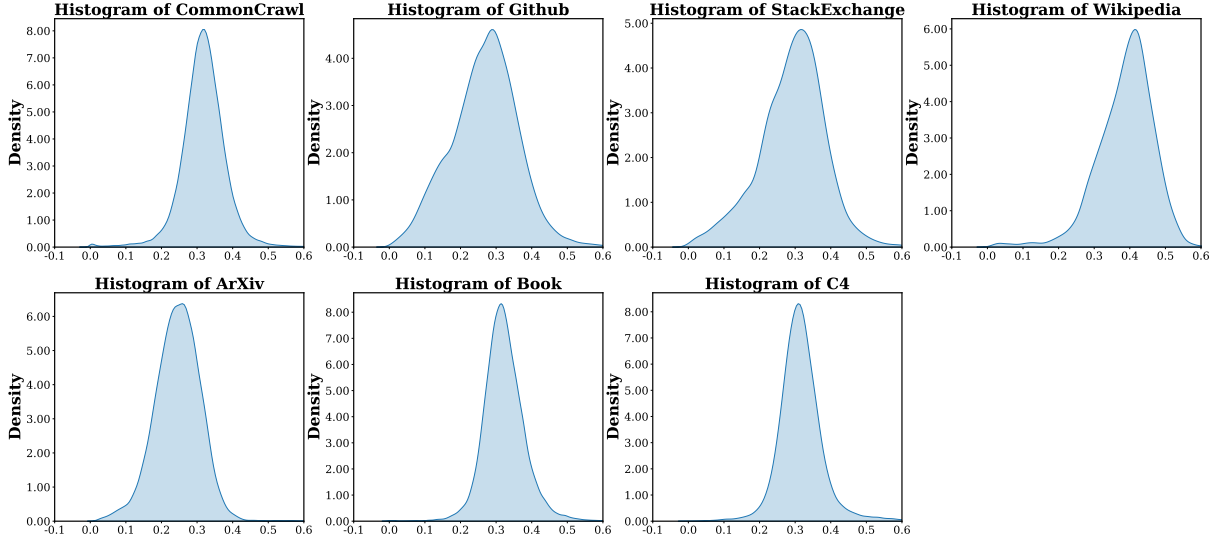Figure 12: PPL distribution across different sources.

Figure 13: PD Distribution across different sources.

B.6 Additional evaluation results

In Figure 15, we present additional evaluation results on 3B models. It is evident that in the majority of benchmarks, FRAME significantly outperforms Random, underscoring the effectiveness of our proposed method.

B.7 Smoother LOSS Curve from FRAME

To analyze the training loss curve's smoothness, we use spectral analysis. We perform a Fast Fourier Transform (FFT) on the loss curve $l[n]$ as $Y[k] = \text{FFT}(l[n])$ to convert it to the frequency domain and compute the Power Spectral Density $PSD[k] = \frac{|Y[k]|^2}{N}$ to examine energy distribution across frequencies. By selecting a cutoff frequency f_c, we divide the spectrum into low and high fre-

quencies and calculate the high-frequency energy proportion $R = \frac{\sum_{f > f_c} PSD[k]}{\sum_{k=0}^{N/2} PSD[k]}$. A smaller R indicates a smoother temporal curve with fewer high-frequency components.

Our analysis reveals that the loss curve from FRAME has the lowest high-frequency energy proportion at 0.02%, significantly lower than that of PDPC and Random. This suggests that FRAME allows the model to converge more stably, and reduces the impact of gradient fluctuations during training. This improvement is due to our four-stage training strategy, which organizes similar data into consecutive batches, thus stabilizing the model training process.

(a) Q_1

(b) Q_2

(c) Q_3

(d) Q_4
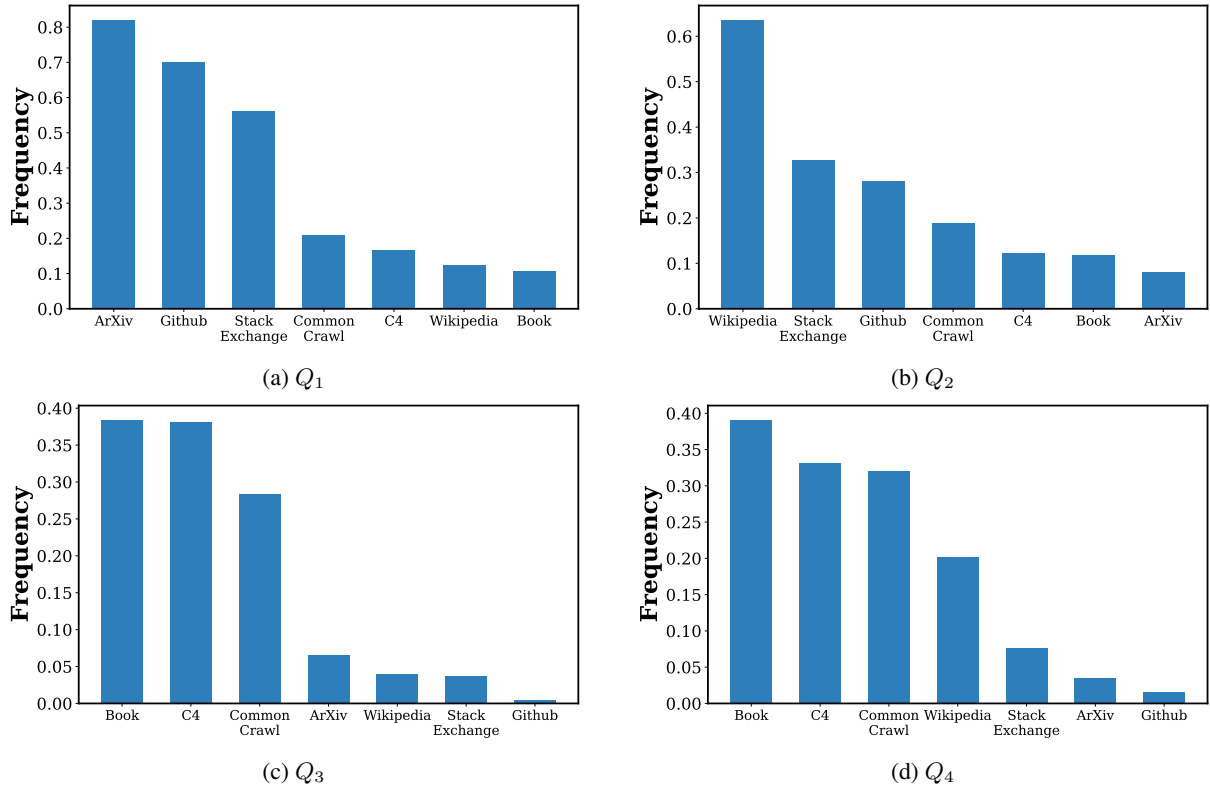
Figure 14: Data distribution details. We've rearranged the order based on the frequency of different sources appearing in each quadrant.

B.8 Ablation Studies

In Figures 16, 17, 18 and 19, we provide a more detailed presentation of the performance of our ablation experiments across various subsets.

PPL Ordering During two-stage training, we observed that training in the order from high PPL to low PPL enhances the model's emergent capabilities. This training strategy shows significant advantages across multiple datasets, such as MMLU and CMMLU. Following the training path from Q_3 to Q_1 can significantly improve accuracy on benchmarks. However, if the opposite training order is adopted (i.e., from Q_1 to Q_3 or from Q_2 to Q_4), the model's performance is similar to that of a randomly initialized model, and in some cases, even worse. This indicates that the impact of training order on model performance is asymmetric, highlighting the importance of properly arranging the training stages.

In three-stage training, the same trend persists. Specifically, after completing training from Q_1 to Q_2 and then switching to Q_4, following the order of data PPL from small to large, the model's accuracy on multiple benchmarks is even lower than that of Random.

PD Ordering From the perspective of two-stage training, the training order from small PD to large PD is beneficial for model training. We found that whether in the same PPL region (e.g., Q_3 to Q_4 or Q_1 to Q_2) or between different PPL regions (e.g., Q_3 to Q_2), the model can ultimately achieve performance superior to that of Random.

This conclusion is also applicable in three-stage training. During the transition from the second stage (Q_1) to the third stage (Q_2) (Figure 19), the model's final accuracy still improved. This further supports the effectiveness of the small PD to large PD order, emphasizing the importance of properly arranging training steps in multi-stage training.
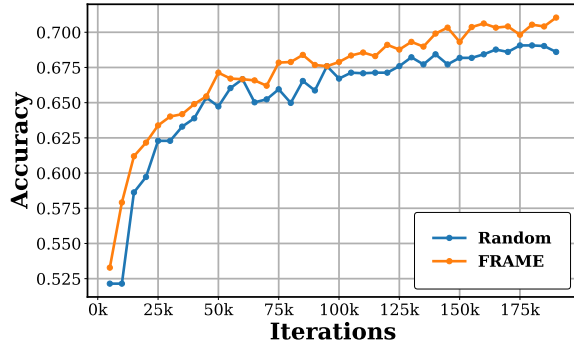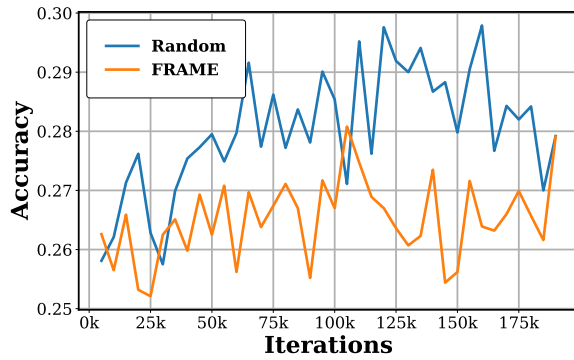
(a) ARC-E
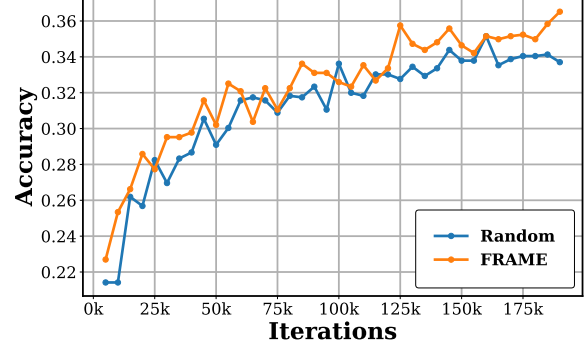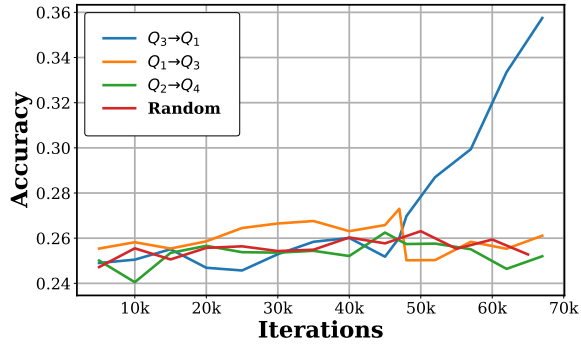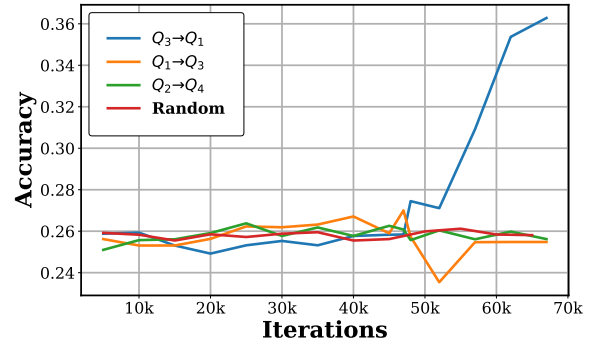
(b) ARC-C
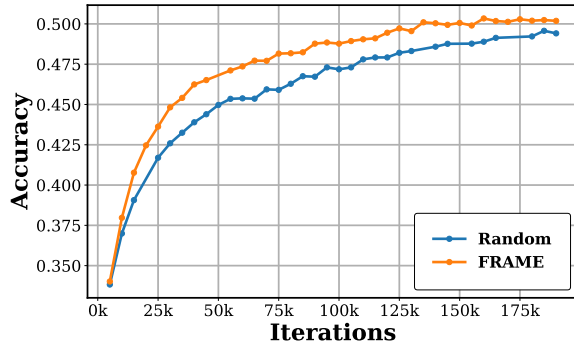
(c) HellaSwag
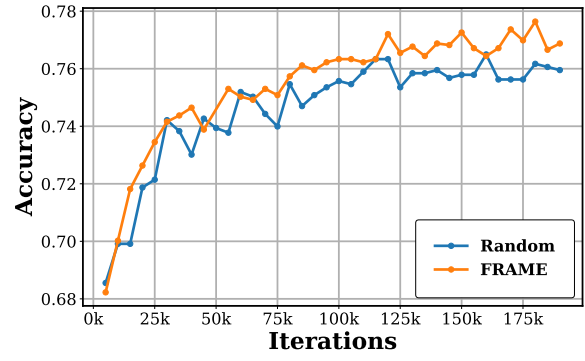
(d) PIQA

(e) BBH

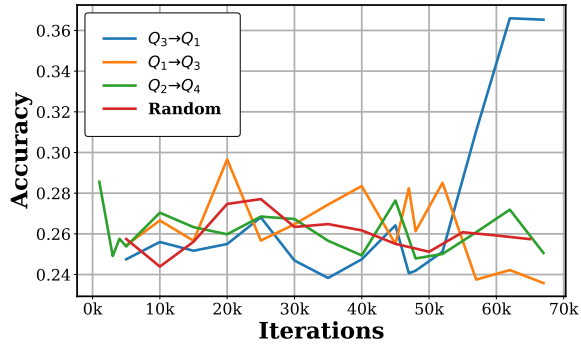Figure 15: Performance of downstream tasks on 3B models with respect to training iterations. We compare FRAME with Random baseline.
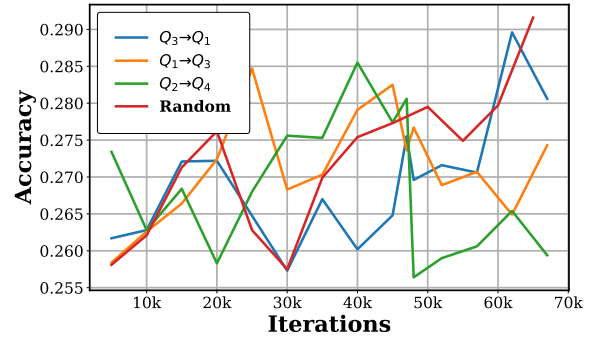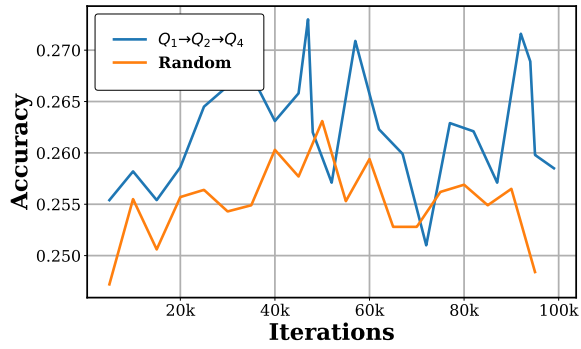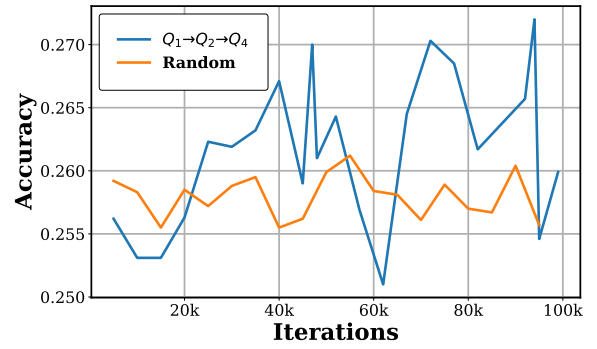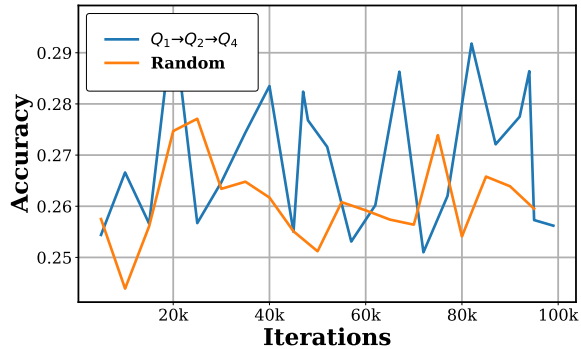
(a) MMLU

(b) CMMLU

(c) CEVAL

(d) BBH

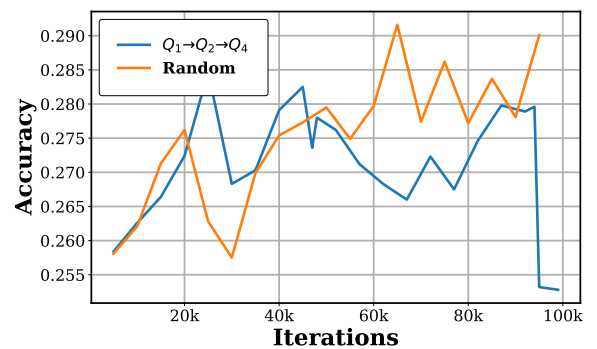Figure 16: PPL related ablation study (2 quadrants).

(a) MMLU

(b) CMMLU
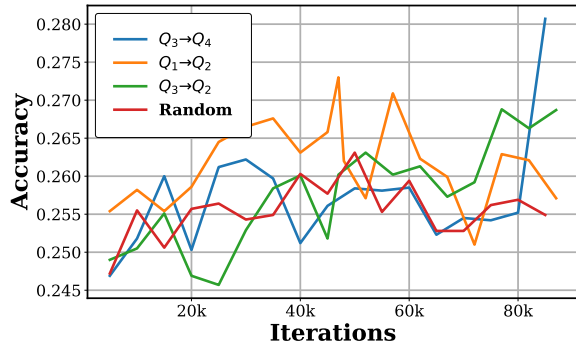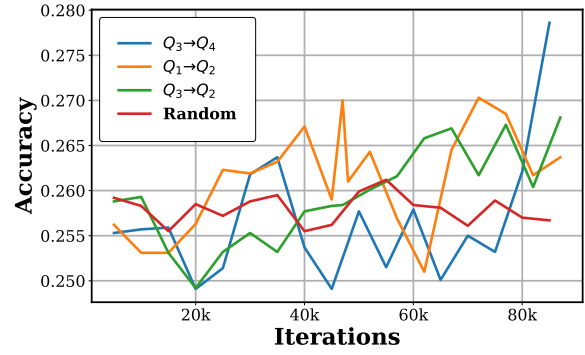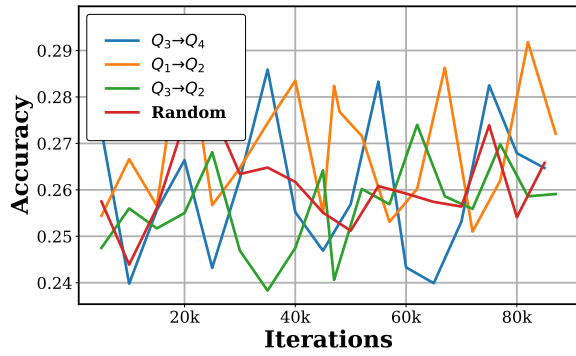
(c) CEVAL

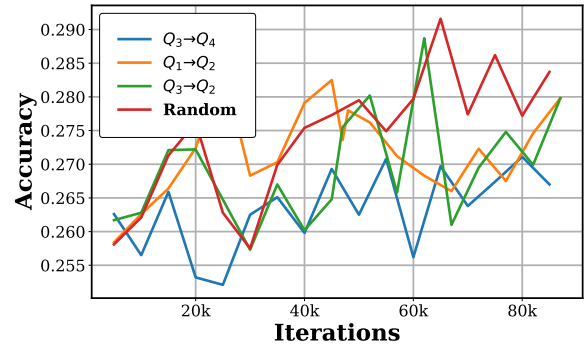(d) BBH

Figure 17: PPL related ablation study (3 quadrants).
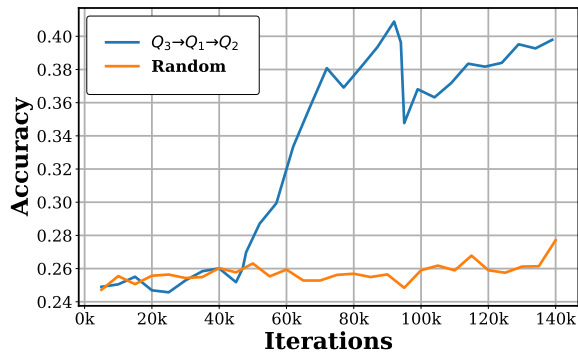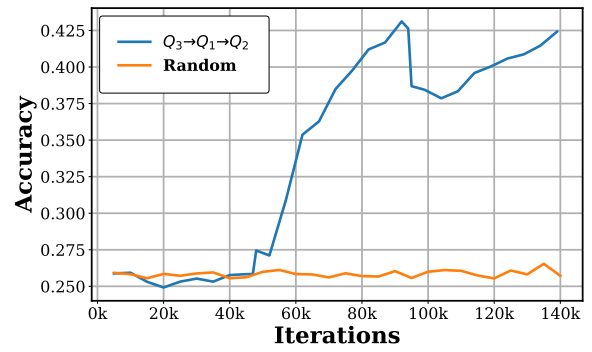
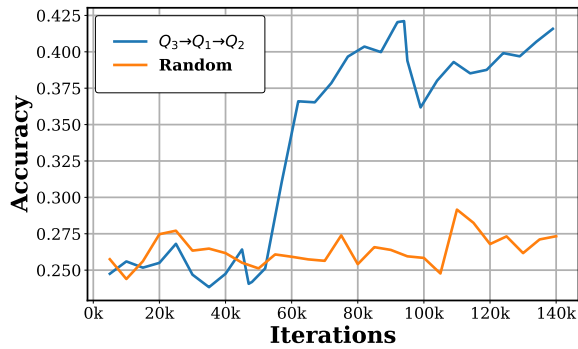(a) MMLU

(b) CMMLU

(c) CEVAL

(d) BBH

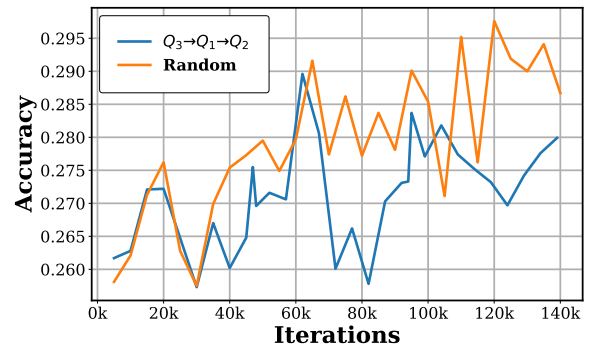Figure 18: PD related ablation study (2 quadrants).

(a) MMLU

(b) CMMLU

(c) CEVAL

(d) BBH

Figure 19: PD related ablation study (3 quadrants).

C Prompts for Case Study

The prompt used in Section 3.4 to analyze the linguistic features of data across different qhadrants is as follows.

Prompts for Property Recognition

You are a language model training data annotator. Your task is to identify whether the given text possesses the following characteristic: {Property}

The text to be annotated is:
{text}

Please determine whether the given text possesses this characteristic according to the above rules.

The output format should be "Because..., my answer is 'X'." where X must be either "yes" or "no ."

You should remain objective and refrain from adding any further comments after making your choice.

The property comes from the following rules:

1. Does the text contain polysemous words? Polysemous words may make understanding more difficult.

2. Does the text use specialized terminology? Specialized terminology may require specific domain knowledge to understand.

3. Does understanding the text require specific cultural background knowledge? Cultural background dependence may increase the complexity of understanding.

4. Does the text require logical reasoning to understand? Logical reasoning adds depth to understanding.

5. Does the text contain elements of humor? Humor may affect the way the text is understood.

6. Does the text explore ethical or moral issues? This may increase the depth of thought.

7. Does the text use complex sentence structures? Complex sentence structures may increase the difficulty of understanding.

8. Does the text contain scientific or technical concepts? These concepts may require specific knowledge to understand.

9. Does the text express obvious emotional tones? Emotional tones may affect the understanding of the text.

10. Does understanding the text require additional background knowledge? Background knowledge requirements may affect the comprehensibility of the text.

D Data Cases

Table 9 presents samples extracted from each quadrant.

Quadrant 1

Sample 1: ... 112. Ra4+ Kd5 113. Ra5+ Kc4 114. Ra6 Qd7+ 115. Kb6 Kb4 116. Ra7 Qd6+ 117. Kb7 Kb5 118. Kc8 Qf8+ 119. Kb7 Qd8 120. Ra1 Qe7+ 121. Kc8 Qe8+ 122. Kb7 Qe4+ 123. Kc8 Qc4+ 124. Kb7 Qf7+ 125. Kc8 Qg8+ 0-1 Analysis of Utegaliyev - Goh from the Baku Olympiad - Life is tough! ***Addendum: The Rook & Bishop vs Rook ending at move 102 was covered specifically by Karsen Muller & Frank Lamprecht on page 301 of Fundamental Chess Endings. My annotations are slightly confusing there ...

Sample 2: ... Women With an Alcoholic Parent Have More Risk Factors Ray Kachatorian / Photographer's Choice / Getty Images There are differences in how parental alcoholism affects daughters as opposed to how it affects sons, particularly when it comes to psychopathology, or mental health disorders, in each gender. Daughters of alcoholics are affected by a parent's alcoholism in many of the same ways that sons are. Both are at higher risk of developing alcohol abuse disorders compared to children of non-alcoholic parents. But there are some differences in how women are influenced, scientists say ...

Quadrant 2

Sample 1: ... pension advice be sought in the case of South Africa. Residents in South Africa are subject to tax (up to 40%) on their total income no matter the source-country. Foreign pensions are exempt from this. Any UK pension, as long as it was not 'received or accrued' from or in South Africa, and is 'in consideration of past employment' elsewhere, is exempt from income tax. This specifically excludes anyone whose pension has arisen from time in public office in South Africa. For residents of South Africa, ...

Sample 2: ... , Bengaluru (As Prepared for Delivery) Honorable President Pranab Mukherjee, Ministers, Excellencies, Distinguished guests: Thank you, it is such a great honor to be here today, accepting this award among such distinguished company. Let me first thank His Excellency the Honorable President Mukherjee, Prime Minister Modi, and the Government of India2013 2026 By U.S. Mission India | 9 January, 2017 | Topics: Chennai, Press Releases, Speeches | Tags: Bilateral relationship Remarks by Ambassador Richard R. Verma at the inaugural plenary of Indo-Asia Connectivity for Shared Prosperity ...

Quadrant 3

Sample 1: ... information may be obtained from CPO. The Anti-Kickback Act of 1986 (41.U.S.C.51-58) was passed to deter subcontractors from making payments and contractors from accepting payments for the purpose of improperly obtaining or rewarding favorable treatment in connection with a prime contract or subcontract. Imposes criminal penalties on any person who knowingly and willfully engages in the prohibited conduct addressed in paragraph (a) of this subsection. Provides for the recovery of civil penalties by the United States from any person who knowingly engages in such prohibited conduct and from any person whose employee, subcontractor, or subcontractor employee provides, accepts, or charges a kickback ...

Sample 2: ... also wrote Brave New World and numerous essays. Consider some of his words as follows: We are in a process of developing a whole series of techniques which will enable the controlling oligarchy to get people to actually love their servitude. A really efficient totalitarian state would be one in which the all-powerful executive of political bosses and their army of managers control a population of slaves who do not have to be coerced, because they love their servitude. To make them love it is the task assigned, in present-day totalitarian states, to ministries of propaganda ...

Quadrant 4

Sample 1: ... in which 3000 people died in a single atrocity, to one's horror at the deaths of ten and perhaps sometimes twenty times as many in each of the bombings of such places as Hamburg, Dresden, Tokyo, Hiroshima and Nagasaki. This way of grasping the purport of what area bombing meant, really meant, is vital to making a difference to how we behave and what we accept today in the conduct of conflicts. There is nothing abstract or theoretical about the mass murder in which bombing consists: it is real and terrible, and anything that drives the point home has its place in the debate ...

Sample 2: ... and China in Africa. Ba'ath Party Dominance and Mistakes. Iraq. A Fragile Country. Burkina Faso. The Church under attack. DR Congo. Insecurity and bad governance contribute to the spread of Ebola. Drug Trafficking. RD.Congo. Beatification of twenty martyr missionaries on track. Africa. Start-ups rolling out. Between Maras and Pandilla El Salvador. A Country In Transition. Nigeria. Troubles in Kano. A New President. The Church of Africa. A Return To Its Origins. Mexico. Until dignity becomes the custom. Philippine. Modern Day Missionaries of the World. The Gumuz People, their culture. The Religious Universe of the Gumuz. The Gumuz in front of a changing world. Africa. The art of food. Ancient flavours and genuine ingredients ...

Table 9: Samples from different quadrants.