Preference Curriculum: LLMs Should Always Be Pretrained on Their Preferred Data

Xuemiao Zhang[1,4*], **Liangyu Xu**[4*], **Feiyu Duan**[2,4*],
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) generally utilize a consistent data distribution throughout the pretraining process. However, as the model's capability improves, it is intuitive that its data preferences dynamically change, indicating the need for pretraining with different data at various training stages. To achieve it, we propose the **P**erplexity **D**ifference (PD) based **P**reference **C**urriculum learning (**PDPC**) framework, which always perceives and uses the data preferred by LLMs to train and boost them. First, we introduce the PD metric to quantify the difference in how challenging a sample is for weak versus strong models. Samples with high PD are more challenging for weak models to learn and are more suitable to be arranged in the later stage of pretraining. Second, we propose the preference function to approximate and predict the data preference of the LLM at any training step, so as to complete the arrangement of the dataset offline and ensure continuous training without interruption. Experimental results on 1.3B and 3B models demonstrate that PDPC significantly surpasses baselines. Notably, the 3B model trained on 1T tokens achieves an increased average accuracy of over **8.1%** across MMLU and CMMLU.

1 Introduction

Large language models (LLMs) have shown impressive performance on various tasks after being pretrained on vast amounts of data (Touvron et al., 2023; Dubey et al., 2024; Liu et al., 2024). As LLMs undergo extensive pretraining, their capabilities steadily improve, which influences their performance and data preferences (Yu et al., 2024). Existing methods of uniformly sampling data throughout the pretraining process are suboptimal because they overlook the model's evolving data preferences (Wettig et al., 2024; Abbas et al., 2023; Sachdeva et al., 2024).

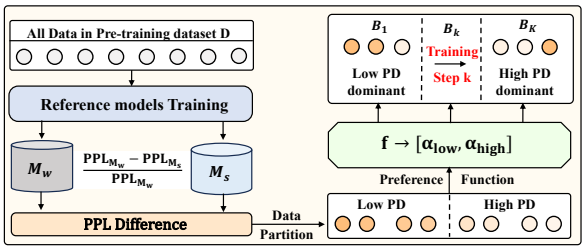


Figure 1: PD-based Preference Curriculum Framework.

Recently, some research has shifted the focus to data influence on model capability during pretraining (Evans et al., 2024; Yu et al., 2024; Koh and Liang, 2017; Ko et al., 2024). A typical method named MATES considers the changing data influence on models but requires interrupting the training process to select more preferred data based on the model's current state, which disrupts training continuity and stability (Yu et al., 2024). A similar issue arises in JEST (Evans et al., 2024).

In this paper, we introduce perplexity (Ziegel et al., 1976) difference (PD) to quantify the difference in how challenging a sample is for LLMs' final and early checkpoints and use PD to gain a deeper understanding of model characteristics. We further propose a novel **PD**-based **P**reference **C**urriculum learning framework, PDPC, as shown in Figure 1. It perceives LLMs' data preference at any training step and uses the preferred data to continuously pretrain LLMs without interruption, thereby boosting their performance. To this end, PDPC needs to solve three major challenges:

How to dynamically perceive preferences without interrupting pretraining. Ideally, training would be paused at any training step to calculate PD using the model's current state, allowing for data sampling from the preference distribution like MATES (Yu et al., 2024). However, to ensure con-

*Equal contribution.
†Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2025, pages 21181–21198

tinuity and stability in pretraining, we propose an offline processing method to approximate the ideal dynamic preference adjustment. First, we calculate PD for all samples offline using both the fully trained checkpoint and an early checkpoint of the experimental model. Then, we develop a preference function to predict the model's preference for data with specific PD characteristics, enabling the pre-organization of data according to the model's preferences at various training stages. Because the data is fully arranged offline, the training of the experimental model proceeds without interruption.

Calculating PD for the entire dataset is prohibitively expensive. Typically, the sizes of experimental models and the pretraining dataset are quite large. Training an experimental model using the entire dataset in a random sampling setting is very costly. Moreover, calculating PD values offline necessitates inferring the entire dataset twice—using both early and final checkpoints—which is extremely costly. Essentially, the differences between various model states can be captured by the disparity in trained FLOPs: the early checkpoint corresponds to fewer trained FLOPs, whereas the final checkpoint corresponds to more. Naturally, we can approximate the FLOPs difference by using two reference models (RMs) with fewer parameters compared to the experimental model, pretraining them on the same data scale. The smaller and larger RM play the roles of the early checkpoint and the final checkpoint, respectively. This approach significantly reduces both pretraining and inference costs.

How to use PD to orderly arrange pretraining data. Data with high PD are challenging for weak models but well-suited for strong models due to the increased capacity. Conversely, data with low PD are less sensitive to model capability differences and can be understood by both strong and weak models. From the perspective of FLOPs, the early stages of training an experimental model can be seen as involving a weak, smaller model, while later stages involve a strong, larger model. Thus, placing high-PD data in the later training stages can enhance the model's ability to fit these challenging samples, whereas low-PD data can be trained earlier since they are less sensitive to model capability. This establishes a natural curriculum learning principle for the pretraining: begin with low-PD data and progress to high-PD data. Following the principle, we propose an S-shape function that effectively models preferences throughout training. Further, to maintain diversity in each batch, we use the concentration of low-PD data rather than PD itself as the output of the preference function.

By overcoming the challenges, PDPC can always perceive and use the data preferred by LLMs to pretrain and boost them. Notably, PDPC only arranges the given data without performing data selection. In summary, our work has the following contributions: (1) We propose PD to measure the difference in the fit of strong and weak models to samples, pointing out that high-PD data is challenging for weak models and is suitable to be arranged in the later pretraining stage. (2) We propose a novel PDPC framework that always perceives and uses the data preferred by LLMs to train and boost them, and ensures uninterrupted continuous training, which serves as the last data preprocessing step. (3) Experimental results show significant performance improvements over baselines. Notably, the 3B model trained on 1T tokens with PDPC demonstrates an average accuracy increase of **4.1%** across all benchmarks and **8.1%** across MMLU and CMMLU.

2 PD-based Preference Curriculum

In this section, we propose PDPC, which always perceives and uses the data preferred by LLMs to pretrain and boost them. Notably, PDPC only organizes the given data without performing selection, which can serve as the final data preprocessing step before pretraining.

2.1 Problem Formulation

Given a pretraining dataset $\mathcal{D}$ following a uniform distribution, we aim to arrange it into a form that better aligns with the oracle distribution $\mathcal{O}_{\{B_k\}}$, which represents the ideal data distribution preferred by the model. To achieve this, we aim to find the optimal preference function f^* that adjusts the joint distribution of all batches $\{B_k\}_{k=1}^{K}$ to closely approximate $\mathcal{O}_{\{B_k\}}$.

$$\begin{aligned} f^* = \arg\min_{f} \ & \text{Divergence}\left(\mathcal{P}_{\{B_k\}}, \mathcal{O}_{\{B_k\}}\right) \\ \text{s.t.} \quad & \bigcup_{k}^{K} B_k = \mathcal{D},\ B_k = f\left(\frac{k}{K}\right), \end{aligned} \tag{1}$$

where $\mathcal{P}_{\{B_k\}}$ represents the joint distribution of the batches, and K is the total number of batches.

2.2 PD-based Data Partitioning

Perplexity Difference (PD). We begin by introducing the concept of PD. Consider two models, the weak model M_w and the strong model M_s, both trained on an identical dataset $\mathcal{D}$. Given a sample x, the PD is defined as:

$$PD(x) = \frac{PPL_{M_w}(x) - PPL_{M_s}(x)}{PPL_{M_w}(x)},$$
$$PPL_{M_*}(x) = exp(-\frac{1}{L_x}\sum_{t=1}^{L_x}\log P(x_t|x_{<t})), \tag{2}$$

where $PPL_{M_w}(x)$ and $PPL_{M_s}(x)$ are the perplexity values of the sample x calculated using M_w and M_s, respectively. L_x denotes the token length of the sample x, and x_t represents the t-th token. "*" indicates weak or strong.

PD indicates the extent to which the strong model outperforms the weak model on a given sample. Low PD indicates that the strong model M_s and the weak model M_w have a similar level of fit to the sample x. Conversely, a high PD value suggests that M_s outperforms M_w in fitting the sample x, implying that the sample is more challenging for M_w to learn. There are very few samples whose perplexity in M_w is smaller than in M_s, accounting for less than 0.01%, and we ignore them.

Data Partitioning. An intuitive method for data arrangement is to sort the pretraining data by PD from low to high. However, this method is suboptimal. As shown in Figure 8 and 9 of Section 3.5, data with extremely high and extremely low PD values differ significantly. Sorting by PD creates batches with overly homogeneous samples, lacking the diversity essential for pretraining LLMs (Sachdeva et al., 2024).

To solve this issue, we propose to partition data into distinct parts based on mutually exclusive PD ranges. During each pretraining step, data from each part is mixed in varying proportions. Formally, given a pretraining dataset $\mathcal{D}$, we calculate the PD for each data point x, denoted as PD_x. We then sort $\mathcal{D}$ in ascending order based on PD_x and partition it evenly into n parts, $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n$:

$$|\mathcal{D}_i| = \frac{|\mathcal{D}|}{n}, \quad \forall i \in \{1, 2, \ldots, n\}, \tag{3}$$

For any $i < j$, the condition holds that:

$$\forall x \in \mathcal{D}_i, \forall y \in \mathcal{D}_j, \quad \text{PD}_x \leq \text{PD}_y, \tag{4}$$

2.3 Definition of the Preference Function

We introduce a preference function $f(p)$ to capture the model's preferences for different PD partitions at any training step. The function maps the pretraining progress p to a proportion vector $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_n]$, where α_i denotes the fraction of data from the i-th domain in the current batch. The training completion rate p is defined as: $p = \frac{k}{K}$, where k is the current training step and K is the total number of training steps. The function $f(p)$ is defined as

$$f(p) \rightarrow [\alpha_1(p), \alpha_2(p), \ldots, \alpha_n(p)],$$
$$\text{s.t.} \sum_{i=1}^{n} \alpha_i(p) = 1, \tag{5}$$

which determines how the proportion of each part changes as training progresses.

To ensure full utilization of data from all parts by the end of the training, we establish the constraint $\int_0^1 \alpha_i(p)\, dp = \frac{1}{n}$ for all $i \in \{1, 2, \ldots, n\}$. In each training step, the current proportion vector $\boldsymbol{\alpha}$ guides the proportional selection of samples from each part $\mathcal{D}_i$. These selected samples are then combined to form a new mixed batch, which is used for the training step.

2.4 Exploration of the Preference Function

Following the CL principle discussed in Section 1, **PDPC starts the training process using low-PD data and gradually moves to high-PD data.** To implement this, we sort the pretraining data and partition them into n parts, allowing us to explore the model's preference for mixing ratios of data with different PD values at various pretraining steps. We discuss different scenarios:

(1) When $n = 1$, the training process involves randomly selecting individual samples from the dataset for each training step.

(2) When $n = |\mathcal{D}|$, each part corresponds to a single sample, which is equivalent to performing a full sample-level sorting of all the data.

(3) When $1 < n < |\mathcal{D}|$, each training step will include data from different parts. Finding the optimal function $f^\star$ is a complex problem that involves substantial costs.

We focus on the case where $n = 2$ because it effectively captures the essential differences between low-PD and high-PD data while remaining computationally manageable. Essentially, $n = 2$ partitions the data into high-PD and low-PD parts.

To guide the offline organization of pretraining data, We introduce a PD-based preference function $f(\frac{k}{K})$, which predicts the proportion of low PD data that the model prefers at different training steps k. However, directly optimizing the preference function f is challenging. To solve this issue, we propose a function search method[1] to approximate f^*. To ensure equal volumes of high and low PD data and align with the model's data preferences, it is crucial to choose the right function. This function ensures that the proportion b of low PD data matches the pretraining completion $p = \frac{k}{K}$, expressed as $b = f(p)$, and must meet specific criteria:

Firstly, based on the CL principle, the function $f(p)$ should exhibit a decreasing trend, gradually increasing the proportion of data with high PD to raise the curriculum difficulty.

Secondly, to ensure that the total amount of data with high PD and low PD stays equal, the function can be symmetric about the point $(0.5, 0.5)$, satisfying $f(0.5 + \Delta) = 1 - f(0.5 - \Delta)$, where $\Delta \in (0, 0.5)$.

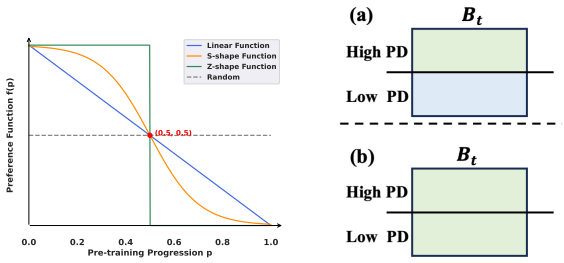


Figure 2: Left: Preference Functions and Right: a comparison of different data sampling methods, the regions highlighted in green represent the preferred data.

We initially search for three representative functions as candidates: the linear function $f_L(p)$, the Z-shape function $f_Z(p)$, and the S-shape function $f_S(p)$, as shown in the left part of Figure 2. $f_L(p)$ represents a steady decline, while $f_Z(p)$ indicates a sudden, distinct change. To introduce a balance between the two, we also incorporate an S-shape function form.

$$f_L(p) = k \cdot (p - 0.5) + 0.5, \tag{6}$$

$$f_Z(p) = \begin{cases} 1 - \lambda, & \text{if } p < 0.5 \\ \lambda, & \text{if } p \geq 0.5, \end{cases} \tag{7}$$

[1]We also discuss an annealing-based iterative optimization approach in Appendix B to approximate f^*.

Algorithm 1 PD-based Preference Curriculum Learning

Input: dataset $\mathcal{D}$, total iterations K, batch size N
Output: trained model θ_K
Initialize model parameters θ_0
Train RMs on i.i.d. subset of $\mathcal{D}$
Calculate PD: for all samples in D using RMs
Partition $\mathcal{D}$ into 2 sub-domains A_{PD}^{low} and A_{PD}^{high}.
Explore and determine the form of preference function:
    $f(p) = \frac{1}{1+exp(a(p-0.5))}$
for $k = 0$ **to** $K - 1$ **do**
    Calculate pretraining progress $p = \frac{k}{K}$
    Get the proportions vector:
        $[\alpha_1, \alpha_2] \leftarrow [f(p), 1 - f(p)]$
    Sample from the two domains to form B_k:
    $B_k = \{x \mid x \sim A_{PD}^{low}\}_{\alpha_1 \cdot N} \cup \{x \mid x \sim A_{PD}^{high}\}_{\alpha_2 \cdot N}$
    Train the model on B_k and update θ_k
end for

$$f_S(p) = \frac{1}{1 + exp(a(p - 0.5))}, \tag{8}$$

where $k \in [-1, 0)$, $\lambda \in [0, 0.5)$, a modulates the steepness of the curve.

Essentially, assuming that at step i the model prefers high PD data, traditional methods (Figure 2(a)) dilute the benefits of high PD data with less favorable low PD data, similar to average pooling. In contrast, PDPC (Figure 2(b)) clusters preferred data within each batch, resembling the effect of Max-Pooling on this data.

3 Experiments

3.1 Experimental Setup

General Setting We train two experimental models: a 1.3B model using 100B randomly selected tokens from the SlimPajama dataset (Soboleva et al., 2023), and a 3B model on a bilingual dataset containing 1T tokens, which comprises 500B tokens each of Chinese and English data, sourced from domains such as books(Gao et al., 2020), blogs(Baumgartner et al., 2020), patents(Sharma et al., 2019), Common Crawl(Penedo et al., 2024), and Wikipedia, similar to the Matrix dataset(Zhang et al., 2024a). We train 100M and 700M reference models (RMs) on an i.i.d. subset with 50B tokens from SlimPajama for the 1.3B setting, and 100M and 1.3B RMs on an i.i.d. subset with 500B tokens for the 3B setting, respectively. All models were trained using the Llama architecture(Touvron et al., 2023) within the Megatron framework (Shoeybi et al., 2019), utilizing the Adam optimizer. We set the batch size to 640 and the context window length to 8192. The initial learning rate is set to 2e-4, with a warm-up phase of over 375M tokens. We adopt a cosine learning rate schedule and set weight decay

to 0.1. A full pretraining run of the 3B model on 1 trillion tokens, utilizing 512 Ascend 910B NPUs, requires approximately 180 hours.

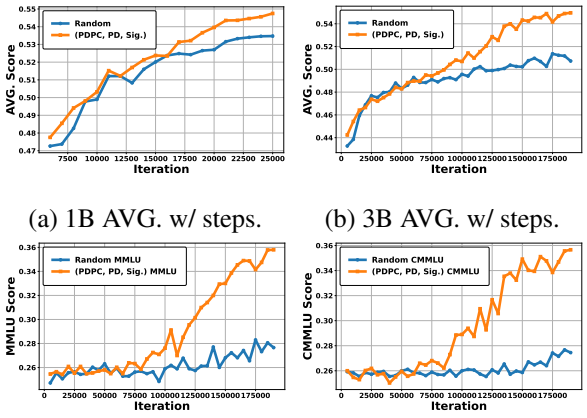
Evaluation We employ the lm-evaluation-harness (Gao et al., 2021) to measure the models' performance on the following benchmarks: ARC-E (Clark et al., 2018), ARC-C (Clark et al., 2018), SciQ (Welbl et al., 2017), HellaSwag (Zellers et al., 2019) and PIQA (Bisk et al., 2020), which include tasks like knowledge question answering and commonsense reasoning. For the 3B model, we add benchmarks like MMLU (Hendrycks et al., 2020), CMMLU(Li et al., 2024), and CEVAL(Huang et al., 2023), which cover multi-domain knowledge and complex reasoning tasks, presenting challenges absent in 1.3B models. We employ in-context learning for evaluation following QuRating (Wettig et al., 2024). Standard accuracy is used as the final metric for all tasks.

Baselines We compare PDPC with *Random* and several basic curriculum learning approaches: **(1)** ***Random***: Each batch is randomly selected from the entire dataset, corresponding to the case of $n = 1$ in our framework. **(2)** ***PPL***: PPL directly measures how well a model fits the data. We use two 700M models, each trained separately on the SlimPajama and Matrix subsets, to annotate their corresponding data. **(3)** ***QuRating*** (Wettig et al., 2024): We select the Education Value in QuRating as the difficulty indicator for curriculum learning. **(4)** ***Sequential***: We fully sort the data based on PD, PPL, and Qurating, arranging them in either ascending or descending order.

3.2 Main Results

Table 1 and 2 present our primary experimental results, revealing several key insights:

Effectiveness of our PDPC framework. PDPC with $n = 2$ consistently outperforms all baselines, regardless of the metric used, showing significant performance and convergence improvements over the baseline. Notably, the 3B model trained on 1T tokens with PDPC demonstrates an average accuracy increase of **4.1%** across all benchmarks and **8.1%** across MMLU and CMMLU, highlighting the effectiveness of our framework. Figure 3 depicts performance improvements in the 1.3B and 3B models as training progresses, with our method significantly outperforming *Random* in the latter half of pretraining. In this phase, data dominated by high PD is crucial, especially in the 3B model, highlighting the effectiveness of transitioning from low to high PD data, which significantly boosts model performance and promotes emergent capabilities.



(a) 1B AVG. w/ steps. (b) 3B AVG. w/ steps.

(c) 3B MMLU w/ steps. (d) 3B CMMLU w/ steps.

Figure 3: Few-shot downstream performance with respect to training steps for Random and (PDPC, PD, S.).

The *Sequential* method can somewhat constrain model performance. Sorting pretraining data by PD from low to high *Sequential-PD-Low2High* can outperform *Random* but falls short of the *PDPC-PD-S.* strategy. This limitation arises because sorting strictly by PD reduces data diversity, leading to homogeneity that restricts the model's ability to handle complex tasks. In extreme cases, if the dataset contains duplicate samples, a complete sort would likely place identical samples in the same batch, which is detrimental to improving pretraining efficiency.

PD performs better than other metrics in the Preference CL framework. Comparison of the results from *Preference CL-PPL-S.*, *Preference CL-Qu.Edu-S.*, and *PDPC-PD-S.* shows that using PD as a metric yields the best results, particularly on the ARC-C and SciQ datasets. PD can accurately reflect the relative difficulty and complexity of samples, which aligns well with the CL principle discussed in Section 1. In contrast, relying solely on PPL or educational value may not effectively capture the differences in sample difficulty required for this CL principle.

3.3 Ablation Study

Impact of PD calculation methods We focus on two main factors: **(1) Size of the RM**: We use PD calculated from various model combinations. **(2)**

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** ↑2.9	**87.9** ↑2.1	33.7 ↓0.5	**68.0** ↑0.7	**54.7** ↑1.2

Table 1: Downstream tasks results on **1.3B** models with **100B** tokens. We report accuracy for each task, and the best performances are marked in bold. Abbreviations: HellaSw. = HellaSwag, AVG. = Average, S.=S-shape Function, S.R.=S-shape Reverse Function.

Method	Metric	Order	ARC-E	ARC-C	SciQ	PIQA	MMLU	CMMLU	CEVAL	AVG.
-	-	Random	68.6	33.7	94.6	76.0	27.7	27.5	27.2	50.8
PDPC	PD	S.	**69.7** ↑1.1	**35.8** ↑2.1	**95.3** ↑0.7	**76.3** ↑0.3	**35.8** ↑8.1	**35.6** ↑8.1	**36.1** ↑8.9	**54.9** ↑4.1

Table 2: Downstream tasks results for different settings after training **3B** models on **1T** tokens.

Choice of the RM: We use early and late checkpoints from a randomly trained 1.3B model to calculate PD and compare it with PD from RMs. The experimental results, as shown in Figure 4, demonstrate that regardless of the scale of the RM or the calculation method chosen, the results consistently outperform *Random*. This validates the robustness of our framework regarding PD calculation methods. Additionally, PD from the 100M-700M RMs slightly outperforms that from the early/end models, further supporting our hypothesis that approximating the early and late checkpoints of the model with RMs is valid, as they are comparable in terms of training FLOPS.

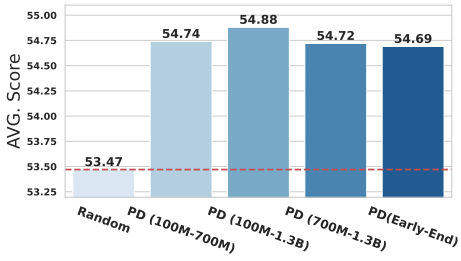


Figure 4: Ablation on PD calculation methods.

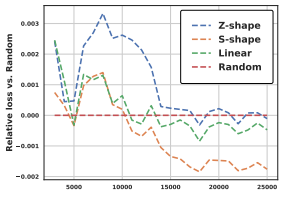
Investigation of different preference functions Table 3 presents the effects of the three preference functions in Section 2.3. The S.R. function is an explored variant of the preference function and is symmetric to the S-shape function with respect to the linear function. Experimental results indicate that the S-shape function outperforms other functions. Its slower initial decline compared to the linear function and S.R. function highlights the importance of starting with enough low-PD data and gradually introducing high-PD data to enhance performance. The Z-shape function, which uses only low-PD data initially and high-PD data later, slightly outperforms *Random*. The linear, Z-shape, and all S-shape parameter settings outperform *Random*, confirming the robustness of our framework.

3.4 Analysis

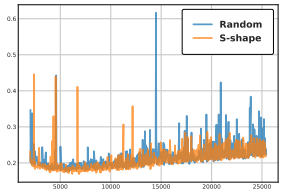
Loss Analysis We sample 500M tokens from SlimPajama as test set to compare test loss with *Random* on the 1.3B setting. Figure 5(a) shows that PDPC's test loss initially declines slowly, then rapidly decreases, achieving a lower loss than *Random* by incorporating higher PD data later in training. The S-shape function effectively helps to minimize loss. Additionally, Figure 5(b) demon-

Function type	H.P.	ARC-E	ARC-C	SciQ	HellaSw.	PIQA	AVG.
Random	-	56.5	23.6	85.8	34.2	67.3	53.5
Z-shape	-	55.6 ↓0.9	23.6 0.0	86.2 ↑0.4	33.8 ↓0.4	**68.9** ↑1.6	53.6 ↑0.1
Linear	-	55.5 ↓1.0	23.9 ↑0.3	87.7 ↑1.9	34.2 0.0	67.3 0.0	53.7 ↑0.2
S-shape	a=2.5	57.1 ↑0.6	24.8 ↑1.2	87.6 ↑1.8	34.0 ↓0.2	67.5 ↑0.2	54.2 ↑0.7
	a=5.0	**57.5** ↑1.0	26.0 ↑2.4	87.7 ↑1.9	**34.3** ↑0.1	67.4 ↓0.1	54.6 ↑1.1
	a=7.5	56.4 ↓0.1	25.4 ↑1.8	87.2 ↑1.4	34.2 0.0	**68.9** ↑1.6	54.4 ↑0.9
	a=10.0	57.3 ↑0.8	**26.6** ↑3.0	**87.9** ↑2.1	33.7 ↓0.5	68.0 ↑0.7	**54.7** ↑1.2

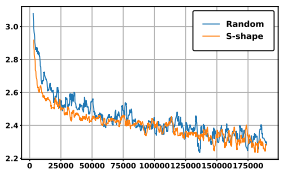
Table 3: Downstream tasks results for different preference functions. We report accuracy for each task, and the best performances are marked in bold. Abbreviations: H.P. = Hyper-parameters.



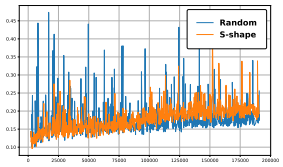
(a) Relative loss vs. Random.



(b) 1B gradient norm w/ steps.



(c) Training loss vs. Random.



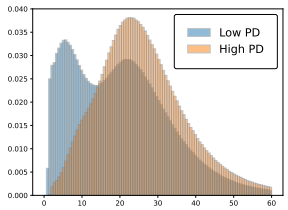
(d) 3B gradient norm w/ steps.

Figure 5: (a) Relative test loss and (b) gradient norm during model training of 1B model. (c)Training loss and (d) gradient norm during 3B model training.

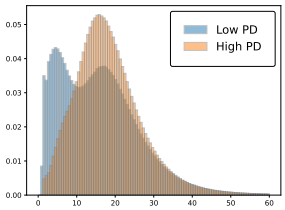
strates that our method stabilizes the gradient norm, ensuring smoother model convergence. In the 3B model experiments (Figure 5(c) and 5(d)), the S-shape function significantly outperforms *Random*. It accelerates early loss reduction, speeds up convergence, and achieves a lower final loss.

PPL Distribution of low-PD and high-PD data As shown in Figure 6, examining the perplexity distribution across data with varying PD values reveals that samples with low PD exhibit lower perplexity. This observation aligns with the trends illustrated in Figure 5c, where training with low-PD data leads to a rapid decrease in training loss during the initial stages, providing the model with a clear direction for gradient optimization. Notably, even when handling with high-PD data in later stages, the model maintains steady loss reduction, ultimately achieving a minimized training loss.

Spearman correlation coefficient of PDs from different RM sizes We evaluate the Spearman correlation coefficients between different PD types,



(a) PPL of 100M RM.



(b) PPL of 700 RM.

Figure 6: PPL distribution of low-PD and high-PD data.

as shown in Figure 7, and find a strong correlation among PDs derived from RMs of varying sizes, which indicates calculating PD with smaller RMs produces results consistent with larger RMs, optimizing computational resources.

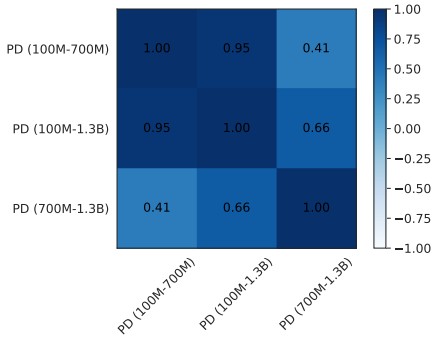
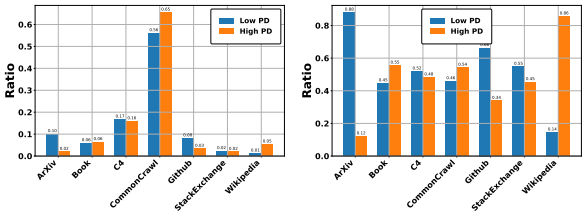


Figure 7: Spearman correlation coefficient of PDs from different RM sizes.

3.5 Case Study

Data Source Distribution Our analysis of 25M texts from Slimpajama (Soboleva et al., 2023) reveals significant differences in distribution and semantics between high-PD and low-PD data. High-PD data mainly comes from Wikipedia and CommonCrawl, while low-PD data is sourced from arXiv and GitHub, as seen in Figure 8. The significant performance improvement of PDPC in later

stages is largely attributed to the higher proportion of high-quality data from sources like Wikipedia, which primarily appears in high-PD data and is less prevalent in low-PD data.



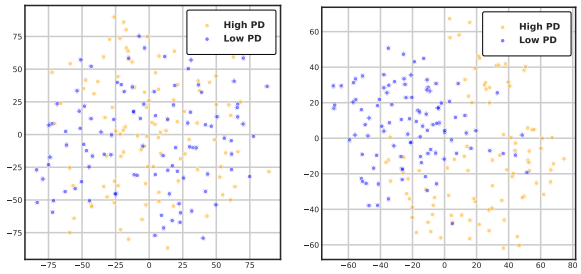
(a) Normalization along domains. (b) Normalization along PD partitions.

Figure 8: Data distribution across different sources.

To understand the semantic structure of the data, we use T5 (Raffel et al., 2023) to generate dense vector representations of texts collected in two ways: (a) uniformly from different PD partitions, and (b) from extreme PD intervals (top/bottom 10%) after sorting. We then apply t-SNE for dimensionality reduction. Figure 9 illustrates the semantic visualization of the data points. Uniform sampling results in high and low PD data being evenly distributed in semantic space, indicating semantic diversity. In contrast, extreme PD sampling leads to distinct semantic spaces, explaining the suboptimality of *Sequential-PD-Low2High*, as it may result in a lack of data diversity in some batches during training, thereby affecting model performance.

Data Quality Distribution We use 4 raters from QuRating(Wettig et al., 2024) to assess data quality. Figure 10 shows consistent quality distributions in both low-PD and high-PD parts, ensuring uniform quality throughout the pretraining process and preventing the model from learning from lower-quality data at any stage.

Stability of PD We evaluate the Spearman correlation coefficients between different PD types (as shown in Figure 7) and find a strong correlation among PDs derived from RMs of varying sizes, which indicates that PD is a relatively stable metric. Calculating PD with smaller RMs yields results consistent with larger RMs, saving computational resources. Furthermore, larger size discrepancies among RMs result in broader PD distributions, which enhance data differentiation (detailed in Figure 12 of Appendix C.2). This finding is supported by ablation tests, which show that PD calculations using models ranging from 100M to



(a) Uniform sampling. (b) Extreme sampling.

Figure 9: Analysis of semantic distributions.

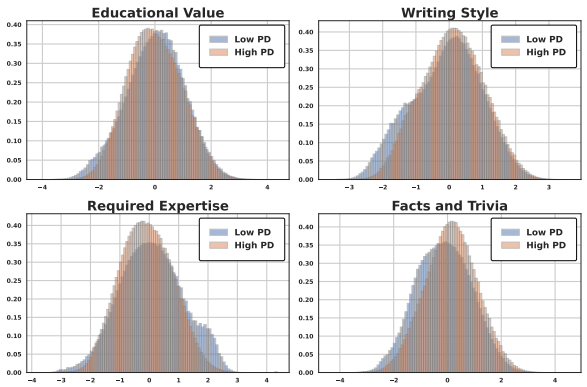


Figure 10: Distribution of low-PD and high-PD data across different quality dimensions.

1.3B yield the best results. Additionally, PD maintains a consistent distribution across domains. For instance, the PD between 100M and 700M models generally appears to follow a normal distribution with an approximate mean of 0.27. Partitioning and organizing data using PD ensures that the data at each training step does not skew towards specific sub-domains, allowing the model to encounter a diverse range of data throughout the entire training process.

Semantic Properties Analysis To explore the semantic features of high-PD and low-PD data, we analyze 1,000 samples randomly sampled from each part using 10 criteria focused on semantic features. These criteria encompass polysemy, specialized terminology, cultural context, logical reasoning, humor, ethical dimensions, intricate sentence structures, scientific concepts, emotional nuances, and background knowledge. Each criterion is clearly defined for GPT-4o to assess with "yes" or "no" responses. More details about the prompt can be found in the Appendix C.4. Figure 11 shows that PD is independent of other linguistic features. Partitioning and organizing data with PD maintains diversity in semantic properties, ensuring that model

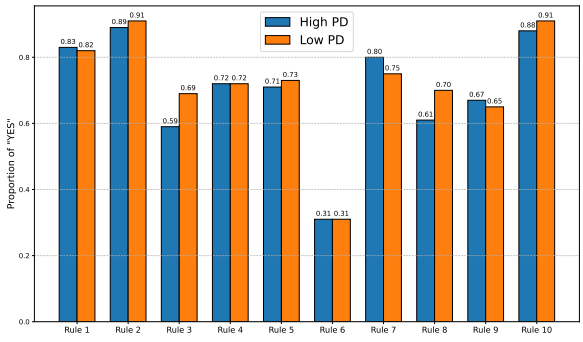


Figure 11: Semantic properties differences of low-PD data and high-PD data.

performance is not restricted by data homogeneity.

4 Related Works

Data preprocessing is crucial in LLM pretraining, ensuring dataset quality and integrity. Traditional methods use expert-crafted rules to filter low-quality data and remove duplicates (Raffel et al., 2020; Rae et al., 2021; Laurençon et al., 2022; Computer, 2023; Penedo et al., 2024; Duan et al., 2025). Enhanced approaches leverage target data sources or proxy models for curation (Wenzek et al., 2020; Xie et al., 2023; Marion et al., 2023). Automated data selection using classifiers is gaining traction; for example, Du et al. (2022) employed logistic regression to evaluate data quality, and other studies have developed sophisticated scoring mechanisms (Zhang et al., 2024b; Sachdeva et al., 2024). QuRating (Wettig et al., 2024) uses multiple raters to assess data contributions. Curriculum Learning (CL) complements these efforts by organizing training data from simple to complex, improving learning efficiency and generalization (Forestier et al., 2022; Soviany et al., 2021). In NLP, CL enhances models, such as in word embeddings (Collobert and Weston, 2008) and neural machine translation (Platanios et al., 2019). Recently, CL's application in LLM pretraining is also growing (Wu et al., 2024).

5 Conclusion

In this paper, we propose PDPC to address the limitations of consistent data distribution in pretraining LLMs. PDPC perceives the models' preferences and utilizes different, model-preferred data as the models' capabilities improve to enhance their performance. We introduce PD as a data metric and incorporate the preference function based on PD to predict data preferences, enabling the offline organization of data and ensuring uninterrupted pretraining. Experiments show that PDPC significantly outperforms the baselines, with the 3B model achieving an average improvement of **8.1**% over *Random* on MMLU and CMMLU.

6 Limitations and Future Works

Exploration of additional PD partitions This study primarily focuses on the scenario where $n = 2$, analyzing concentration mixing curves and systematically blending two subsets with higher and lower PD in accordance with training progression. However, we have not yet explored dividing the training data into more than two subsets to assess whether further performance enhancements are attainable. In future research, we plan to investigate cases where $n > 2$ and develop novel methodologies for addressing learning curves.

Iterative update of learning curves We determine the S-shaped learning curve through functional exploration and use it as the basis for arranging the data sequence to train the model. In fact, we can also start from the newly trained model, re-explore new learning curves, and iteratively update our curriculum learning path. Optimizing the learning curve through multiple iterations could be one of our future research directions.

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*.

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.

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*.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep

neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167.

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*.

Talfan Evans, Nikhil Parthasarathy, Hamza Merzic, and Olivier J. Henaff. 2024. Data curation via joint example selection further accelerates multimodal learning. *Preprint*, arXiv:2406.17711.

Sébastien Forestier, Rémy Portelas, Yoan Mollard, and Pierre-Yves Oudeyer. 2022. Intrinsically motivated goal exploration processes with automatic curriculum learning. *Preprint*, arXiv:1708.02190.

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*.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, et al. 2021. A framework for few-shot language model evaluation. *Version v0. 0.1. Sept*, 10:8–9.

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*.

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.

Myeongseob Ko, Feiyang Kang, Weiyan Shi, Ming Jin, Zhou Yu, and Ruoxi Jia. 2024. The mirrored influence hypothesis: Efficient data influence estimation by harnessing forward passes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26286–26295.

Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR.

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.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2024. Cmmlu: Measuring massive multitask language understanding in chinese. *Preprint*, arXiv:2306.09212.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

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*.

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*.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom M Mitchell. 2019. Competence-based curriculum learning for neural machine translation. *arXiv preprint arXiv:1903.09848*.

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.

Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. 2021. Curriculum self-paced learning for cross-domain object detection. *Preprint*, arXiv:1911.06849.

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*.

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*.

Biao Wu, Fang Meng, and Ling Chen. 2024. Curriculum learning with quality-driven data selection. *arXiv preprint arXiv:2407.00102*.

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.

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*.

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*.

Eric R. Ziegel, G. E. P. Box, G. M. Jenkins, and G. C. Reinsel. 1976. Time series analysis, forecasting, and control. *Journal of Time*, 31(2):238–242.

A Ethical Considerations

Due to the influence of training data, LLMs are prone to generating untruthful or socially harmful content. We aim to mitigate this issue by enhancing the reliability of model training and the model's final performance through the proposed training data adjustment framework. Additionally, training LLMs incurs substantial time and financial costs. Therefore, exploring ways to maximize the efficiency of training data utilization will be key to addressing this problem and can also contribute to reducing global carbon emissions.

B Preliminary Exploration of Iterative Optimization of Preference Functions

When employing a grid search methodology, the size of the solution space scales as n^T, where T represents the number of training steps. Consequently, an increase in the number of parts n results in an exponential expansion of the solution space.

In Section 2.3, we introduce a curriculum learning method that partitions pretraining data into 2 parts and identifies the S-shape preference function through theoretical analysis. This method is simple and efficient. However, in resource-rich scenarios, we also offer a more precise approach to simulate the model's preferences at different pretraining stages, as shown in Algorithm 2. Specifically, after using the discovered preference function to guide

the model's pretraining, we conduct annealing experiments on checkpoints from different stages to explore the model's preferences for data mixing ratios. Based on these preferences, we fit the model's preference function to guide the next round of pretraining. This process is iterated until the model's performance converges.

B.1 Proportion Preference Annealing Experiment

In this subsection, we aim to systematically explore the model's preference for data mixing ratios based on PD at different pretraining stages. This process is crucial for understanding the dynamic changes in model preferences during the pretraining process and provides empirical evidence for optimizing curriculum learning strategies.

Firstly, we construct the dataset required for the annealing experiment. Based on the median of PD values, samples are divided into two parts: low-PD data and high-PD data. We create 11 different annealing datasets where the proportion of low-PD data takes values of 0%, 10%, 20%, ..., 100%. To enhance data diversity, each dataset is supplemented with 30% of samples that share the same distribution as the pretraining data, and the mixed samples remain consistent across all datasets. This design ensures the robustness and comparability of the experimental results.

The annealing experiments are conducted at various stages of model pretraining to evaluate the model's preference for data mixing ratios at different training progressions. We perform experiments on checkpoints from 8 pretraining stages, corresponding to pertraining progress of 0%, 12.5%, 25%, 37.5%, 50%, 62.5%, 75%, 87.5%, and 100%. At each checkpoint, we evaluate the model using all 11 annealing datasets and record the model's performance across different mixing ratios.

By conducting annealing experiments at checkpoints throughout the pretraining stages, we obtain a series of preference data points (p, b), where p represents the pretraining progress, and b denotes the model's preference for the proportion of low-PD data at that stage. Specifically, for each stage p, b is defined as the proportion of low-PD data that optimizes model performance, i.e.,

$$b = \arg\max_{\beta} \mathcal{M}(\beta), \tag{9}$$

where $\mathcal{M}(\beta)$ denotes the comprehensive performance metric of the model on the annealing dataset with the proportion of low-PD data β.

Algorithm 2 Iterative Optimization of Preference Functions

Input: Pretraining dataset $\mathcal{D}$, initial preference function $f(p)$, termination threshold ϵ
Output: Iteratively trained model θ_K
while not converged **do**
  Partition $\mathcal{D}$ into 2 sub-domains A_{PD}^{low} and A_{PD}^{high}.
  Construct the annealing dataset $\mathcal{D}_i$ with the proportion of low-PD data set to β_i:
  $\beta_i \in \{0\%, 10\%, \ldots, 100\%\}$
  for p in $\{0\%, 12.5\%, \ldots, 100\%\}$ **do**
    Retrieve the model checkpoint θ_p
    Evaluate model performance $\mathcal{M}_p(c)$ on $\{\mathcal{D}_i\}_{i=0}^{10}$
    Record preference $b_p = \arg\max_c \mathcal{M}_p(c)$
  end for
  Use PCHIP to fit $b = f(p)$ from data points (p_i, b_i)
  if change in $f(p)$ between iterations $< \epsilon$ **then**
    Break
  end if
  for $k = 0$ **to** $K - 1$ **do**
    Calculate pretraining progress $p = \frac{k}{K}$
    Get the proportions vector:
    $[\alpha_1, \alpha_2] \leftarrow [f(p), 1 - f(p)]$
    Sample from the two domains to form B_k:
    $B_k = \{x \sim A_{PD}^{low}\}_{\alpha_1 \cdot N} \cup \{x \sim A_{PD}^{high}\}_{\alpha_2 \cdot N}$
    Train the model on B_k and update θ_k
  end for
end while

In our study, to fit the changes of preference of LLMs during pretraining, we employ the Piecewise Cubic Hermite Interpolating Polynomial (PCHIP) method. Given experimental data points (p_i, b_i), the PCHIP method constructs local cubic polynomials to ensure monotonicity and smoothness within each interval. Specifically, for each interval $[p_i, p_{i+1}]$, PCHIP generates a cubic polynomial:

$$S_i(p) = a_i(p-p_i)^3 + b_i(p-p_i)^2 + c_i(p-p_i) + d_i, \tag{10}$$

where coefficients a_i, b_i, c_i, d_i are determined by satisfying interpolation conditions, derivative continuity, and monotonicity conditions. These conditions ensure that the fitting curve not only passes through all data points but also maintains monotonicity within each interval, preventing overfitting.

We apply the interpolation function to a uniformly distributed set of points from 0 to 1 to obtain a continuous function curve of concentration preference b as it varies with pretraining progress p. We constrain the values of the fitted curve between 0 and 1. This method effectively captures the trend of model preferences for datasets at different pretraining stages, providing a reliable foundation for subsequent analysis.

B.2 Iterative Curriculum Learning

After completing the proportion preference annealing experiment and successfully fitting the preference function $b = f(p)$, we apply this function to optimize curriculum learning strategies, guiding the pretraining process of the model. Specifically, based on the fitted function $f(p)$, we dynamically determine the optimal proportion b of low-PD data during training according to the current pretraining completion p.

However, it is important to note that the integral of the fitted preference function over the interval $[0, 1]$ may not equal 0.5. This implies that, under this configuration, the amounts of low-PD and high-PD data used may not be equal. To ensure the reasonable utilization of all data, we calculate the integral of the function over $[0, 1]$ to determine the quantile threshold for dataset division. The specific formula is:

$$\int_0^1 f(p)\, dp = \alpha. \tag{11}$$

where α guides to adjust the data allocation ratio.

After pretraining is completed, we conduct the proportion preference annealing experiment again to obtain updated preference data points (p_i, b_i), and refit the preference function $f(p)$ based on these data. This process continues iteratively until the following termination condition is met: the change in the preference function between two consecutive iterations is below a predefined threshold. This method ensures that the model's data preference is precisely adjusted and optimized.

C Experimental Details

C.1 Computational Efficiency Analysis

In our experiments, computational efficiency during the pretraining stage remained consistent across methods. The main differences in computational cost arise from the additional steps required before pre-training, specifically the training of raters or RMs, as well as the associated data labeling procedures. We estimate the computational cost of QuRating (Wettig et al., 2024) and our method based on scaling laws (Rae et al., 2021), as summarized in Table 4.

Method	FLOPs	Major Components
QuRating	2.6×10^{20}	Rater training + full-data inference
PDPC	6.7×10^{20}	RMs training + PPL inference

Table 4: Computational cost before pretraining.

QuRating requires one forward pass and one additional backward pass for each comparison sample during rater training, beyond standard pre-training. The training set consists of 500K samples, each with 512 tokens. After rater training, a single inference pass over the entire 100B-token dataset is performed to generate difficulty scores. According to scaling laws (Rae et al., 2021), the total computational cost is approximately 2.6×10^{20} FLOPs.

PDPC involves training two RMs—of sizes 100M and 1.3B—on a 50B-token subset. Both models then perform PPL inference over the full 100B-token dataset. The total estimated computational cost amounts to approximately 6.7×10^{20} FLOPs. Although PDPC incurs higher computational cost, this additional cost is justified by its substantial gains in downstream performance, as demonstrated in our experiments.

C.2 PD distribution across different domains

As illustrated in Figure 12, large size discrepancies among RMs result in broader PD distributions, which enhance data differentiation. This finding is supported by ablation tests, where the 100M-1.3B PD calculations yielded the best results. Additionally, PD maintains a consistent distribution across domains. This stability makes PD a reliable metric. Organizing training data by PD ensures it does not skew towards specific sub-domains, allowing the model to encounter diverse data at every stage.

C.3 Detailed performance on the benchmarks

In this section, we explore the detailed performance across various benchmarks. Figure 13 illustrates how these metrics evolve during training. We compare the performance of *Random* and *PDPC-PD-S.*, across pretraining iterations.

Our experiments involved training a model with 3 billion parameters on a dataset containing 1 trillion tokens. This large-scale training setup effectively demonstrates our approach's superior performance. The results highlight the gains in accuracy and performance achieved by our method, showcasing its clear advantages over *Random*. We affirm the potential of the *PDPC-PD-S.* methodology in enhancing model performance, particularly when faced with diverse and challenging benchmarks.

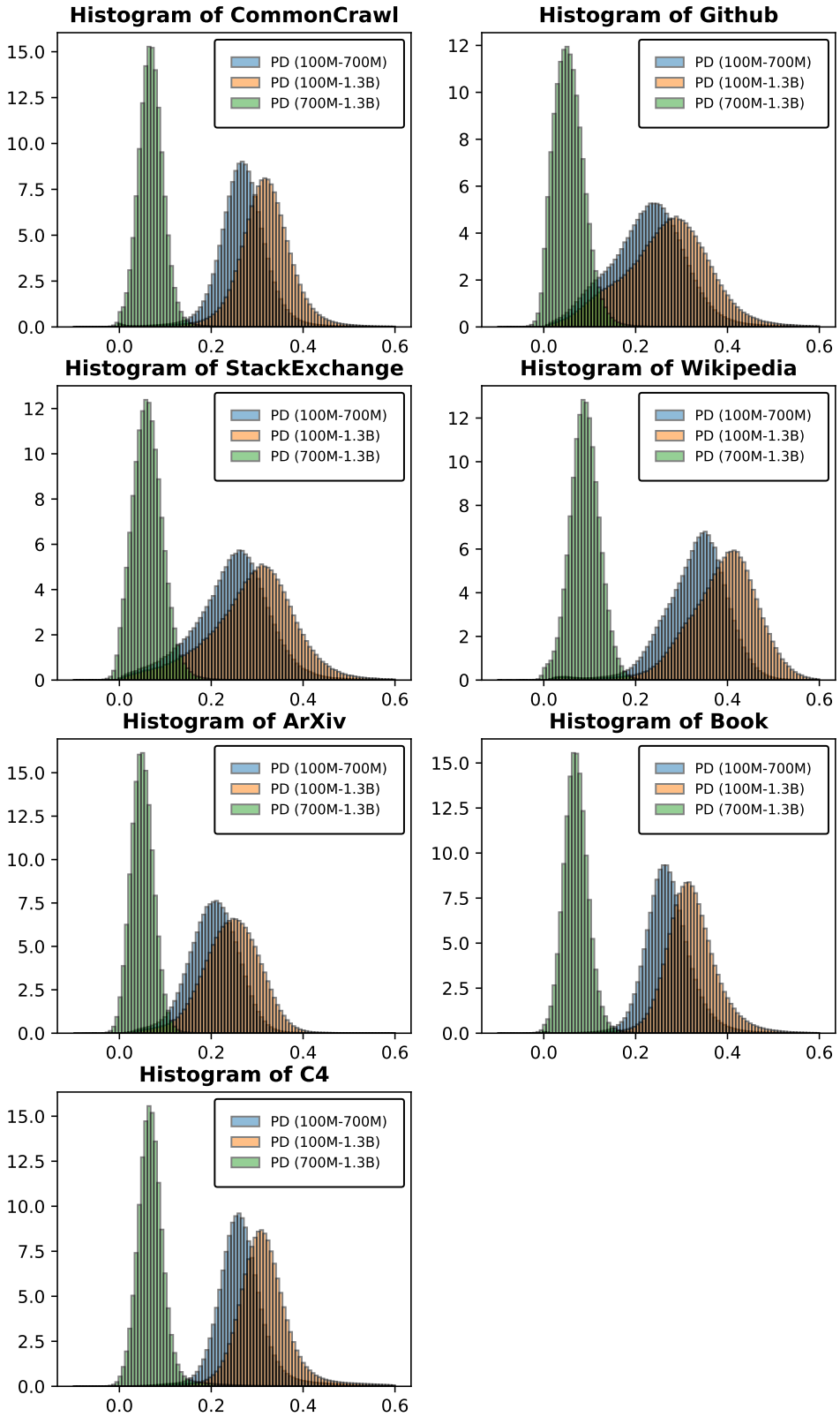
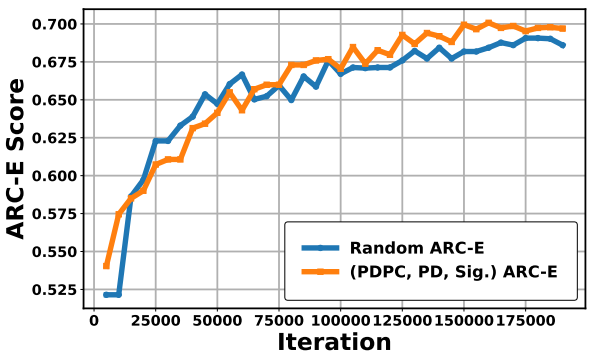
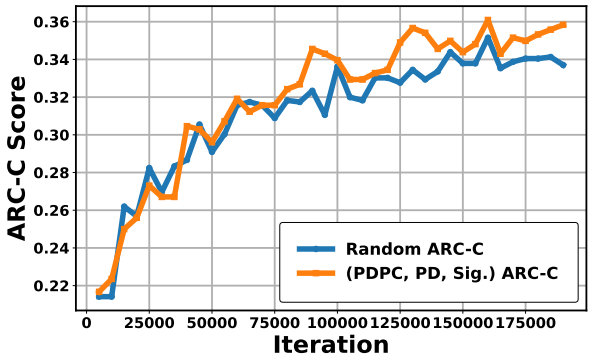


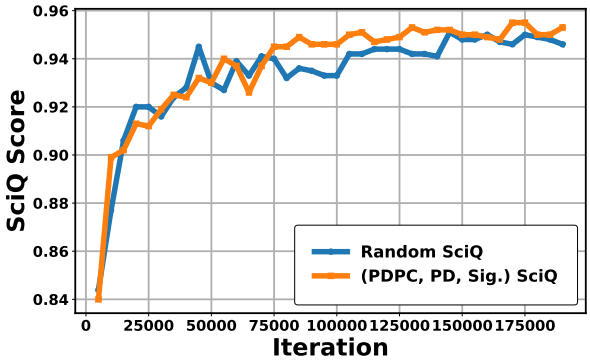
Figure 12: PD distribution across different domains.



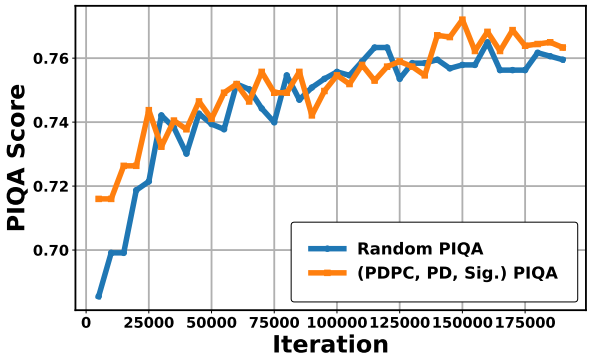
(a) Accuracy on ARC-E.



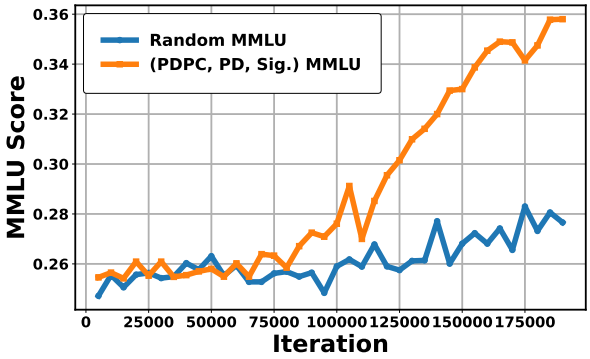
(b) Accuracy on ARC-C.



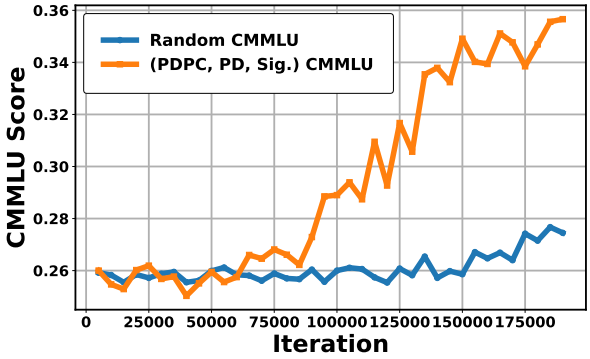
(c) Accuracy on SciQ.



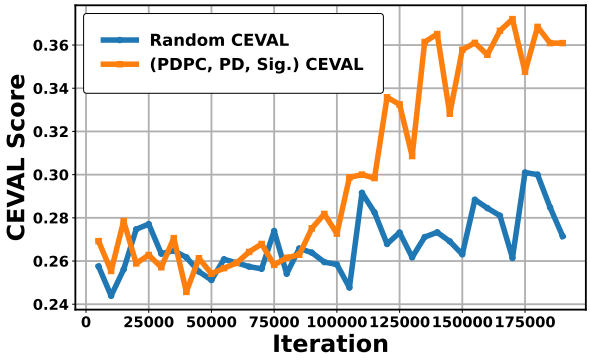
(d) Accuracy on PIQA.



(e) Accuracy on MMLU.



(f) Accuracy on CMMLU.



(g) Accuracy on CEVAL.

Figure 13: Few-shot downstream performance on various benchmarks with respect to pretraining iterations for Random and *PDPC-PD-S.*. We train a 3B model over 1T tokens, demonstrating superior performance with our approach.

C.4 Prompts for Case Study

The prompt and specific rules used in Section 3.5 to analyze the linguistic features of data across different PD intervals are 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.

{Property} is from one of the following rules:

Rules for Property Recognition

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 Case Study

Table 5: Samples are divided into 10 PD quantiles, with two samples representing each quantile.

0-10%
Sample 1: The need to practice good self-care doesn't change in this working environment, but how you accomplish this goal might. Much of Arel's own self-care regimen needed to be adjusted."I was used to weekly massage and monthly chiropractic care. That was gone," she explains. "I am used to runs and yoga and time to meditate in complete silence. That was gone, too." ...
Sample 2: I have to ask you, why'd you–wha–wha–why are you peeing right here?Creepy Guy: What?Kumar: I mean... why'd you pee right next to me when you could like, choose that bush, or–?Creepy Guy: Well, this bush looked like I should pee on it. Why are you peeing on it?Kumar: Well, no one was here when I chose this bush.Creepy Guy: Oh, so you get to pee on it and no one else does? Huh?Kumar: ...
10-20%
Sample 1: boolean insertventas() String sql "INSERT INTO ventas (id_venta, venFechaventa, venId_cliente, venIdadministrador, venTotalventa) VALUES (NULL, '" + vent() "', '" clasu.getId_usuarios() "', '1', '" + pnlProductos.totall + "')"; try con cn.getConnection(); ps = con.prepareStatement(sql); ps.executeUpdate(); return true; catch (SQLException ex) Logger.getLogger(LogicaSql.class...
Sample 2: In terms of providing shorter stay parking, Bell Street multi storey car park is identified as a long stay car park, and the tariffs are so designed to encourage the use of the facility by all day / half day parkers with more flexible tariffs available at other car parks and the on street spaces around the vicinity allow for parking for up to one hour.I have commented that there is no short term (30 minutes to 2 hour) parking available at the West Bell Street multi-storey car park and ...
20-30%
Sample 1: Maybe it just sagslike a heavy load. Or does it explode? by Langston HughesIn 1849, Elizabeth Blackwell became the first woman to graduate from a U.S. medical school in N.Y.In 1864, Rebecca Lee Crumpler became the first black woman to graduate from a U.S. medical school in New England.In 1915, women represented approximately 5% of the physician workforce in the U.S.In 1983, women represented approximately 1/3 of U.S. medical school matriculants ...
Sample 2: FILED UNDER SEAL PURSUANT TO PROTECTIVE ORDER rise to a direct infringement claim against it. See, e.g., Akami Techs., Inc. v. Limelight Networks, Inc., 797 F.3d 1020, 1023 (Fed. Cir. 2015) (noting entities are liable for performance they control). The evidence further shows that Badoo Software Limited and Badoo Limited are also intimately involved in Badoo Trading's creation and ownership of the infringing Bumble application...
30-40%
Sample 1: ... I myself should be a castaway.Young's Literal: but I chastise my body, and bring it into servitude, lest by any means, having preached to others – I myself may become disapproved.As noted earlier, Paul now applies the example from the Greek sports arena directly to himself ("I discipline. . . I myself") and does so that he might present himself as an example or model for other believers to imitate (cp 1Co 4:16, 11:1, 1Th 1:6, cp Heb 6:12, He 13:7, 3Jn 1:11)...
Sample 2: ... (B) of from about 0.1 to about 10.0% w/v of a bioadhesive polymeric stabilizer selected from the group consisting of:(i) polyethylene-polypropylene oxide tri-block co-polymers of the formula;(polyethylene oxide)a -(polypropylene oxide)b -(polyethylene oxide)c wherein PA4 a is 46, 52, 62, 75, 97, 98, 122, or 128; PA4 b is 16, 30, 35, 39, 47, 54, or 67; and PA4 c is 46, 52, 62, 75, 97, 98, 122, or 128;(ii) polyvinyl alcohol,(iii) polyvinyl pyrr.....
40-50%
Sample 1: ... as foreigners seem to have trouble believing about the trees. A second year passed before the fruit split open, and I came out, and several siblings, with hair like Sapham and wings like Pham, and we have no gender because we are not animals but fruit and we like to sing, too, and we like to fly, and we like to be loyal, and we like to love. The tree opened up and flew away and when it was done only twigs and a few blue leaves remained, and then they blew away, too, and we were all born, and ready to live...
Sample 2: ... She had a World Series poker face, and it never slipped. He wondered if she'd had a plan of her own, given how long she went without looking rattled. Maybe she assumed he was in Lenny's corner. Maybe she'd been lining up a double hit.Looking at it now though, it surprised him, how easily he'd committed to killing Lenny. Not that he regretted it, but clearly he was risking fatal penalties, stepping in on a Garcia job and smoking one of their guys. He could live with the risk, but he'd never thought about it at the time...
50-60%

Table 5: (continued)

Sample 1: "We had him just where we wanted him—but it's a fine line." Were they talking about him? They must be. Or was it just arrogance to think that? "I really don't see what has changed," Cheng Li said. "If anything, we're closer to the result we want." Connor felt his head begin to pound. If they were talking about him, what did this mean? Had they had something to do with what had happened to Grace? He remembered in a flash Grace saying that Cheng Li had known her plan...

Sample 2: ...in which, I suspect, the most diverse directions of my work will come together.Added to this is an external incentive. Next year Toledo, I hear, is to be the scene of a big Greco exposition: not only would I like scrupulously to avoid this occasion, I fear that this hitherto still so uninterrupted earthly constellation, which is Toledo, will after this congestion be left changed, popularized, so that this is almost the last moment for surprising it in its remoteness.Now it goes against me, dear friend, to give in to this important decision...

60-70%

Sample 1: ... The stringy, yet short, dark-skinned Mawikizi returned the salute with a smile. "I pulled some serious strings to haul you out here this quick, Keyes." He held the door open for Keyes, and it banged shut behind them once the lieutenant stepped through. "Walk with me." The rough rock-tunneled corridor stretched out in front of them. Mawikizi led Keyes down past offices, shouldering past privates and officers who stood to attention as he walked by. Keyes glanced off down a subcorridor, seeing barracks in the distance...

Sample 2: ... The United States District Court (federal) hires court reporters for its courts, including those within New York State. No test is required. When vacancies occur, announcements for experienced reporters are posted in places such as the NYSCRA website. Reporters who meet stated criteria are told how to apply. Selected candidates are rigorously interviewed for appointment to this important judicial arena. Realtime certification has become a prerequisite for most federal court reporting positions...

70-80%

Sample 1: ... For example, though the Gaon railed against the potentially negative effects of synagogue attendance, it is obvious that women did go to the synagogue. Few shared his jaundiced view, though others did point out the possible pitfalls. Similarly, the Gaon's horror at the prospect of his daughters strolling in the street could not be a guide for the many women who spent their days pursuing their family's livelihood in the marketplace.Beyond the sphere of ritual behavior, a woman was expected to fulfill a religious role analogous to her social function and reflecting her status in society—that is, woman as religious facilitator...

Sample 2: ... Assuming that the amplitude is larger than b_{AC}^{th}, the switching time is determined by the frequency sweeping rate α. Once the magnetic moment is captured intoautoresonance, its nonlinear precession frequency is locked to the instantaneous excitation frequency $f(t) = f_0 + \alpha t$ (remember that $\alpha < 0$).If we define the switching time τ as the time it takes for the moment to cross the energy barrier and knowing that the frequency vanishes atthe top of the barrier. At the top of the energy barrier, the precession reverses from counter-clockwise toclockwise...

80-90%

Sample 1: ... The manipulation and processing of stereo image sequences demand higher costs in memory storage, transmission bandwidth, and computational complexity than of monoscopic images. This chapter investigates scenarios for cost reduction by using reversible watermarking. The basic principle is to embed some data by reversible watermarking instead of either computing or storing/transmitting it. Storage and/or bandwidth are reduced by embedding into one frame of a stereo pair the information needed...

Sample 2: ...Rosengarten pitched the third and fourth innings and Guillozet closed it out. Barhorst finished 2 for 2 at the plate. The Tigers tied in a nonconference game on Monday in Covington. Jackson Center scored two runs in the top of the seventh but Covington scored four runs in the bottom of the inning to tie it. The Tigers scored two runs in the third in three in the fourth, but the Buccaneers scored five in the fifth to tie it 5-5. Jackson Center took the lead with two runs in the sixth. Jackson Center had 10 hits and five errors while Covington had three hits and three errors...

90-100%

Sample 1: ... In fact, it might be fair to say that in this member of an inferior race, there were as many animal characteristics as human ones, but they were gentle and caressing animal ways. He had nothing of the wild animal in him, but rather the physiognomy of a good and faithful dog, like a courageous Newfoundland dog, who can become man's friend and not just his companion. Indeed, he came at the sound of his name, like one of those devoted animals, to rub himself against the master whose hand gripped his own...

Sample 2: ...the whole world has transformed right into a Global City. Details is passed into every corner of the world within minutes. This raising appeal gave rise to numerous information as well as material organizing websites on the net .Wpengine deals pay as you go August 2018 Web holding service is a solution which enables the companies as well as individuals to put information and web content online. . It has many kinds and also groups. Following are its main categories. Wpengine deals pay as you go August 2018 Exactly what is the objective of web organizing...