Towards Optimal Pre-training Data Mixtures for Empowering Reinforcement Learning in LLMs

Xue Han*, **Qian Hu***, **Yitong Wang, Wenchun Gao,**
Lianlian Zhang, Qing Wang, Lijun Mei, Xuefeng Zhao, Junlan Feng†
China Mobile Jiutian Artificial Intelligence Technology (Beijing) Co., Ltd.
{hanxueit, huqianit, wangyitongit, gaowenchun, zhanglianlian,
wangqingit03, meilijun, zhaoxuefeng, fengjunlan}@chinamobile.com

Abstract

Optimal pre-training data mixtures are vital for Large Language Models (LLMs), yet post-RL reasoning performance varies significantly across base models. This suggests that identifying "RL-friendly" pre-training data mixtures is a critical but under-explored prerequisite. This motivates us to ask: Which specific data mixtures in pre-training enhance post-RL effectiveness, and how can we evaluate a base model's suitability for RL fine-tuning? Through large-scale experiments with proxy models, we observe that (1) domain-specific data correlation strongly predicts post-RL performance, and (2) the average perplexity of the base model on k RL-generated responses, avg(k-ppl), is a more stable and correlated predictor than traditional metrics. Building on these insights, we propose BridgeRL, a framework that treats finding optimal mixing ratios as a regression task, using avg(k-ppl) as the fitting objective. We train 1M/60M-parameter proxy models for regression and scale the findings to 1B/3B-parameter models. Results across online and offline RL algorithms demonstrate that BridgeRL-optimized mixtures consistently yield superior base models for RL fine-tuning.

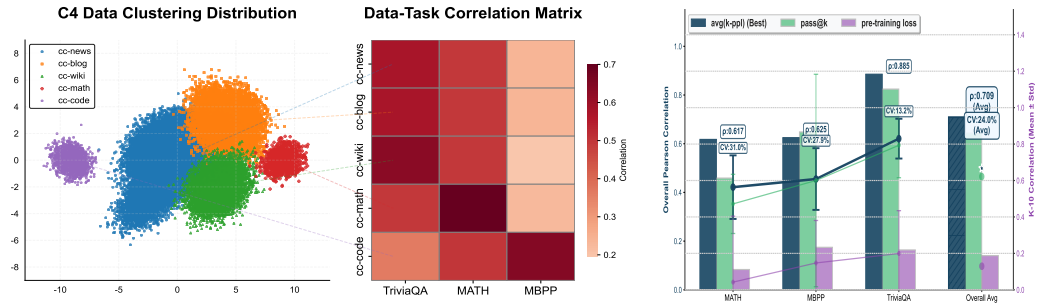
1 Introduction

Optimal pre-training data mixtures are foundational to the emergent reasoning capabilities of Large Language Models (LLMs), as exemplified by the success of OpenAI-o1 [1] and DeepSeek-R1 [2]. Reinforcement Learning (RL) has recently become the de facto standard for further amplifying these capabilities. Although debate persists regarding whether RL truly extends a base LLM's inherent reasoning boundaries [3, 4], there is a consensus that RL efficacy is deeply rooted in the base LLM's initial state. However, a critical performance gap remains: even under identical RL protocols, base LLMs trained on different data mixtures [5, 6, 7, 8] exhibit vastly divergent reasoning efficiencies after RL-finetuning [9, 10]. This unexplained performance variance motivates fundamental questions: **Which specific pre-training data compositions facilitate downstream RL effectiveness, and how can we evaluate a base model's "RL-readiness" without exhaustive full-scale training?**

To address the first question, we investigate the correlation between specific data domains and downstream reasoning performance. We utilize the C4 [11] dataset as it is a high-quality representative of Common Crawl, the primary pre-training resource for modern LLMs [12, 13]. By clustering C4 based on metadata and semantic features, we identify five key sub-domains: cc-news, cc-blog, cc-wiki, cc-code, and cc-math. To quantify the subdomain's impact, we train small proxy models (covering both pre-training and RL phases) using mixtures of these domains at randomly sampled

*Equal contribution.
†Corresponding author.

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



(a) Domain data correlation with downstream performance

(b) Evaluation of surrogate metrics for downstream performance prediction

Figure 1: (a)The code and math datasets exhibit high correlations with their respective downstream tasks. Consequently, our objective is to explore and identify the optimal mixing ratios for how enhancing code and math in pre-training data. (b)We evaluate the predictive efficacy of three candidate metrics—avg(k-ppl), pass@k, and pre-training loss—against downstream task. Dots denote mean Spearman correlation coefficient across k-fold cross-validation. We also calculate the coefficient of variation (CV) for each metric to represents the relative variability. Notably, avg(k-ppl) emerges as the superior indicator: it achieves the highest correlation (ρ, quantifying the association with true performance) and the lowest CV, which is the standard deviation normalized by the mean. These results confirm its superior predictive power and robust cross-validation stability.

ratios. Detailed experimental configurations are provided in Appendix C. As illustrated in the dual-panel Figure 1(a), we evaluate the RL-trained proxy models on TriviaQA [14], MATH [15], and MBPP [16], while simultaneously computing the correlation coefficients between each data category and its respective task accuracy. This analysis reveals the following key insights:

> • ***Domain-Specific Synergy***: *The code and math datasets exhibit strong task-performance correlations, identifying them as pivotal drivers of reasoning capabilities and motivating a search for their optimal mixing ratios. Conversely, since the three internet-sourced datasets yield comparable contributions, we treat these remaining sources as a single 'General' domain without further distinction.*

To address the evaluation question, we maintain consistent RL protocols across both Qwen2.5-3B and Llama2-3B base models, followed by comprehensive assessments on the TriviaQA, MATH, and MBPP benchmarks. We systematically compare the predictive power of three candidate metrics: pass@k (prevailing adopted for evaluating RL-trained LLMs [17]), pre-training loss, and our proposed avg(k-ppl), which represent the average perplexity of the base model calculated over k RL-generated trajectories. We provide more deep analysis in Appendix D.5 and compare with other metrics (RL loss and entropy) in Appendix D.6. As illustrated in Figure 1(b), an analysis of Spearman correlation coefficients reveals that:

> • ***avg(k-ppl) as a Robust Predictor***: *avg(k-ppl) outperforms alternative indicators as a predictive metric, exhibiting both higher correlation coefficients and greater stability (lower variance).*

Leveraging these insights, we introduce BridgeRL, an RL algorithm-agnostic framework that employs avg(k-ppl) as a robust measure of a base model's reasoning potential. Given that the possible mixing ratios for the three domains (general, code, and math) constitute a prohibitively large search space, we formulate the search for optimal pre-training proportions as a regression task [5], utilizing avg(k-ppl) as the target objective. Finally, we employ the fitted regression model to predict the performance of unseen mixtures and implement the optimal configuration to train a large-scale model that achieves better RL performance.

To validate BridgeRL, we follow [5] to train 512 proxy models (both pre-trained and RL-trained) with 1M/60M parameters and scale up to 1B/3B parameters using different data mixtures from three

domains. The experimental results on both representative online (GRPO [18]) and offline (DPO [19]) RL algorithms show that the optimized data mixture predicted by BridgeRL yields a better base model for RL training.

2 Related Works

The relationship between the RL and base model. The success of RL-trained LLMs inspires research interest in exploring their underlying mechanism. Existing studies [20, 3, 10, 17] investigate whether reinforcement learning creates new reasoning solutions from scratch or merely applies existing patterns in base models. They suggest that RL-trained models do not gain new reasoning skills beyond those seen in their base models but rather enhance sampling efficiency along right reasoning paths. On the contrary, ProRL [17] offers surprising new insights that RL can indeed discover genuinely new solution pathways entirely absent in base models when given sufficient training time and applied to novel reasoning tasks. While the prior research has not reached a consensus on whether RL truly expands the reasoning boundaries of base LLMs, they acknowledge that the efficacy of RL is significantly dependent on the initial capabilities of the base model. [9] demonstrates that the reasoning capability of RL-trained LLMs varies among base model families, such as Llama [21] and Qwen [22]. It further proves that the base LLMs could be enhanced for RL through mid-training with a proper data mixture. Differently, we explore how to get an RL-friendly base LLM during the pre-training. [10] seeks to clarify how the composition of pretraining data affects the efficacy of RL fine-tuning using varying proportions of the text and math datasets for pre-training. They focus on analyzing the RL reaction on the pre-training data instead of exploring an optimal pre-training data mixture.

Data mixing. Data mixing strategies significantly influence LLM performance [23, 24]. Existing methods typically employ either offline weighting via proxy models [25, 5] or online dynamic adjustments [26, 6, 27]. While recent works utilize scaling laws or regression to guide large-scale mixing [28, 5], we adopt the regression-based approach [5] due to its flexibility in handling problems lacking direct evaluation metrics. Unlike [5], however, our work focuses on a distinct fitting target tailored for RL-readiness.

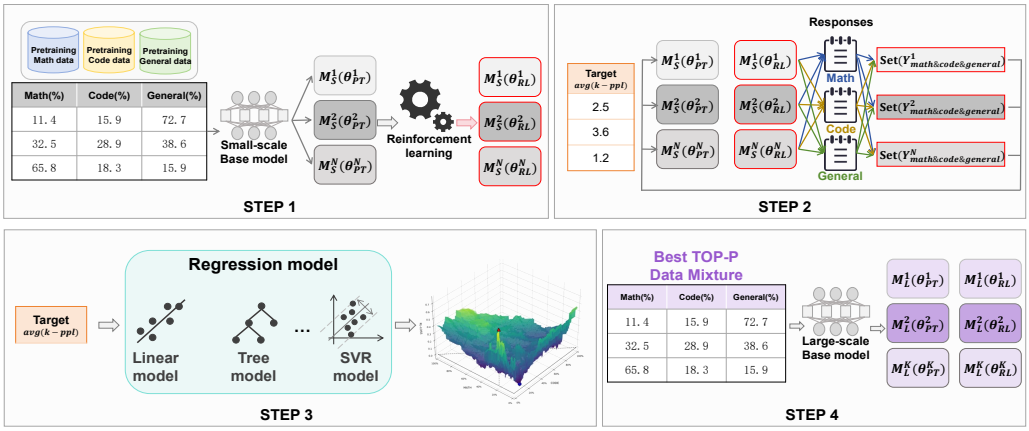


Figure 2: The framework of BridgeRL, including the following steps: 1) Generate random pre-training data mixtures of math, code, and general domains, then pre-train a series of small-scale proxy models based on these mixtures and continually train these proxy models through reinforcement learning. 2) Obtain the target average perplexity avg(k-ppl) of the pre-trained model over k responses generated by the RL-trained model. 3) Fit a regression model using the ratios of mixtures and the avg(k-ppl) as the target label. 4) Leverage the fitted regression model to predict the best mixture for the target value on a large-scale model and further validate the prediction.

3 Methodology

3.1 Overview

The scenario in this paper is to identify the optimal pre-training data mixture ratio that yields the best-performing LLM after both pre-training and reinforcement learning stages. As illustrated in Figure 2, the proposed BridgeRL involves the following four key steps. **(1)** We first train small proxy models, which will be used later for regression. These small proxy models include both pre-trained and RL-trained models. The pre-trained models are trained using numerous different data mixtures from three domains (general corpus, code, and math). It should be noted that our proposed framework is easy to scale to more domains. Following the training paradigm of DeepSeek-R1 [2], our RL-trained models are optimized directly from the pre-trained checkpoints, bypassing the conventional intermediate supervised fine-tuning (SFT) phase. **(2)** With the small proxy pre-trained and RL-trained models, we calculate the average perplexity scores of pre-trained models on k responses generated by RL-trained models, regarded as avg(k-ppl). **(3)** Fit a regression model using the data mixture ratios as features and the avg(k-ppl) calculated in step 2 as the target label of the regression. **(4)** We use the fitted regression model to predict the optimal data mixture on large-scale LLMs and then train models to validate. Next, we will elaborate on the specific details of each step.

3.2 Train small-scale proxy models with random data mixtures

First, we adopt numerous diverse data mixtures to obtain a series of small-scale proxy models, including both the pre-trained base models and RL-trained models. The mixing ratios are sampled based on a token-distribution-based Dirichlet distribution, ensuring broad coverage of weight extremes while preventing overemphasis on low-frequency token domains.

Specifically, leveraging the Dirichlet distribution [29], we randomly generate N different data mixture ratios across three training domains (math, code, and general) to construct a wide spectrum of sparse and near-uniform distributions:

$$\begin{aligned} Set(R) &= [R^1, R^2, ..., R^n, ..., R^N] \\ R^n &= [r^n_{general} : r^n_{code} : r^n_{math}] \end{aligned} \tag{1}$$

$R^n \in Set(R)$ denotes the possible pre-training data mixture ratio of three domains (e.g., $R^n = [6 : 2 : 2]$).

A set of N small-scale proxy pre-trained models $Set(M_S(\theta_{PT}))$ could be obtained, with each M_S pre-trained by adopting each data mixture strategy $R^n \in Set(R)$:

$$Set(M_S(\theta_{PT})) = [M^1_S(\theta^1_{PT}), M^2_S(\theta^2_{PT}), ..., M^N_S(\theta^N_{PT})] \tag{2}$$

With $Set(M_S(\theta_{PT}))$, we then directly apply reinforcement learning with the same setting to train each pre-trained proxy model, ultimately obtaining a series of RL-optimized small proxy models:

$$Set(M_S(\theta_{RL})) = [M^1_S(\theta^1_{RL}), M^2_S(\theta^2_{RL}), ..., M^N_S(\theta^N_{RL})] \tag{3}$$

3.3 Calculate the average perplexity avg (k-ppl)

The pipeline for RL-trained LLMs consists of two phases: pre-training and RL training. Furthermore, RL is a sophisticated and unstable approach that has multiple variations, including both online and offline algorithms with distinct techniques such as DPO [19] and GRPO [18]. As a result, there are no straightforward evaluation metrics for analyzing the data mixture effect on RL-trained models via base models, like the predict-next-token loss function used to evaluate base LLM data mixing.

Drawing inspiration from the pass@k metric widely adopted for evaluating RL-trained LLMs [2, 17, 10], and grounded in our empirical findings (Figure 1(b)), we introduce avg(k-ppl). This metric leverages the average perplexity of the base LLM across k RL-generated responses to quantify the boundaries of the model's inherent reasoning capabilities. Such avg(k-ppl) evaluation is agnostic to the specific RL algorithm utilized and may be adapted to any variety (e.g., offline DPO and online GRPO), enhancing its versatility.

Generally, given a base proxy model $M_S^n(\theta_{PT}^n)$, its corresponding RL-trained proxy model $M_S^n(\theta_{RL}^n)$, an input x, and a response $Y = (y_1, y_2, ..., y_T)$ that is generated by the $M_S^n(\theta_{RL}^n)$, the perplexity of the pre-trained model for Y is defined as the exponentiated average negative log-likelihood of a sequence:

$$\begin{aligned} &PPL_{M_S^n(\theta_{PT}^n)}(Y|x) \\ =&exp(-\frac{1}{T}\sum_{t=1}^{T} logP(y_t|(x, y_1, ..., y_{t-1}))) \end{aligned} \tag{4}$$

Eq.4 reflects the model's ability to predict the given response Y conditioned on the prompt x. Lower perplexity indicates that the model has a higher likelihood of generating this response.

We randomly sample a set of prompts in three domains: $x \in X = [Set(X_{code}), Set(X_{math}), Set(X_{general})]$. Similar to the pass@k [17] metrics for evaluating RL performance, we use each $M_S^n(\theta_{RL}^n)$ to generate k multi-responses for each prompt x. The set of all these responses is denoted as $Set_{RL}^n(Y)$. Let $\overline{k-ppl}_{PT}^n(Y_{RL})$ be $M_S^n(\theta_{PT}^n)$'s average perplexity of these responses generated by $M_S^n(\theta_{RL}^n)$, defined as below:

$$\begin{aligned} &\overline{k-ppl}_{PT}^n(Y_{RL}) = \frac{\sum_{Y_\alpha \in Set_{RL}^n(Y)} PPL_{M_S^n(\theta_{PT}^n)}(Y_\alpha|x_\alpha)}{l_1 + l_2 + l_3} \\ &PPL_{M_S} = [\overline{k-ppl}_{PT}^1(Y_{RL}), ..., \overline{k-ppl}_{PT}^N(Y_{RL})] \\ &Set_{RL}^n(Y) = Set(Y_{code}^{l_1}) \cup Set(Y_{math}^{l_2}) \cup Set(Y_{general}^{l_3}) \end{aligned} \tag{5}$$

Where l_1, l_2, l_3 respectively represent the number of $M_S^n(\theta_{RL}^n)$'s responses in three domains. Y_α denotes each response in $Set_{RL}^n(Y)$, while x_α is its corresponding prompt.

The definition of avg(k-ppl) can effectively evaluate the reasoning capabilities boundaries of pretrained models [17]. Once we have obtained these avg(k-ppl), we can use the data mixture as features and the avg(k-ppl) as target labels to fit a regression model.

3.4 Fit a regression model using the data mixtures as features and the avg(k-ppl) as the target label

The next step is to fit a regression model using the data mixture ratios of three domains as features and the avg(k-ppl) as target labels. We choose the regression model [5] over the scaling law [30] to forecast appropriate data mixture ratios for large-scale LLMs due to a lack of straightforward evaluation metrics for our problem. With the regression model, we are able to leverage a limited set of small models to estimate continuous target values for unknown data mixture ratios.

Specifically, considering the input features correspond to the domain ratios of the data mixture $R^n = [r_{general}^n : r_{code}^n : r_{math}^n]$, and the target label $\overline{k-ppl}_{PT}^n(Y_{RL}) \in PPL_{M_S}$, the goal is to find a function that best maps the input features to the target variable.

Taking linear regression as an example, which assumes a linear relationship between the input features and the target variable, can be represented as:

$$\overline{k-ppl}_{PT}^n(Y_{RL}) = \omega_0 + \omega_1 r_{general}^n + \omega_2 r_{code}^n + \omega_3 r_{math}^n \tag{6}$$

Where ω_0 represents the error or noise in the data, and $\omega = (\omega_1, \omega_2, \omega_3)$ are the coefficients associated with the respective input features $(r_{general}^n : r_{code}^n : r_{math}^n)$. The coefficients ω are typically estimated using techniques like ordinary least squares, aiming to minimize the sum of squared residuals between the predicted and actual target values.

Alternative regression algorithms such as LightGBM [31] or SVR [32] can also be used for fitting the regression models. Using known training data mixtures, these regression models can learn a predictive function that can estimate target values for arbitrary data mixtures without requiring additional training.

Once the regression model is fitted, we can efficiently explore the entire space of possible data mixtures. By leveraging the regression model to predict target values for each potential mixture, we rapidly identify the optimal input features $R^{best} = [r^{best}_{general} : r^{best}_{code} : r^{best}_{math}]$ that yield the lowest $\overline{k-ppl}^{n}_{PT}(Y_{RL})$ within an infinite set PPL_{M_S}:

$$[r^{best}_{general} : r^{best}_{code} : r^{best}_{math}] = \underset{R^n \in Set(R)}{argmin} \ (\overline{k-ppl}^{n}_{PT}(Y_{RL})) \tag{7}$$

This simulation-based optimization significantly reduces costs, as both small-scale model simulations and regression predictions are computationally inexpensive.

3.5 Train large-scale models using the predicted optimal data mixture

After simulating and predicting the optimal data mixtures via the regression model, we apply the predicted top-P mixture ratios to sample large-scale pre-training data:

$$Set(R^{topP}) = [R^1_{top}, ..., R^P_{top}] \tag{8}$$

According to the hypothesis proposed and proved by [5], the rank invariance of data mixtures posits that the relative ranking of data mixtures in terms of their impact on model performance remains consistent across different model scales and training token quantities. Thus, we train a larger model using these optimally mixed data, going through both pre-training and RL phases:

$$\begin{aligned} Set(M^{topP}_L(\theta_{PT})) &= [M^1_L(\theta^1_{PT}), ..., M^P_L(\theta^P_{PT})] \\ Set(M^{topP}_L(\theta_{RL})) &= [M^1_L(\theta^1_{RL}), ..., M^P_L(\theta^P_{RL})] \end{aligned} \tag{9}$$

We can subsequently employ these well-trained larger models to perform the corresponding downstream tasks.

4 Experiments

4.1 Experimental setups

Datasets and benchmarks. For the pre-training stage, we train on a mixture of three datasets for code, math, and general corpus separately: TheStack [33], OpenWebMath [34], and the subset C4 dataset of redpajama [35]. During the reinforcement learning fine-tuning phase, we train on instruction datasets such as python-code-25k [36], OMNI-MATH [37] and GSM8K [38] to further optimize the model. Then, we choose the following representative downstream benchmarks used by LLMs to assess the model's broad generalization capacity. General tasks: OpenBookQA(OBQA)[39], PIQA[40] and Hellaswag[41]. Coding tasks: HumanEval [42] and MBPP [16]. Math tasks: MATH [15] and GSM8K [38]. The adoption of these datasets and benchmarks is aligned with the publicly released LLMs' technical reports [21, 43, 2].

Metrics. For the regression model, we follow [5] and employ the Spearman rank correlation coefficient (ρ) as the evaluation metric. The Spearman correlation ρ provides a nonparametric measure of the strength and direction of association between two variables. For the RL-trained LLMs, each downstream task is evaluated using its corresponding standard evaluation metrics.

Baselines. We compare BridgeRL with six baselines: *Human*, *PPL*, *RL_loss*, *avg(k-entropy)*, *pass@k*, and *Worst*. *Human* employs manual ratio allocation for data across the three domains based on empirical training experience according to the release of LLMs' technical reports [21, 43]. Specifically, the mixing ratio for code, math, and general is 0.33:0.15:0.52. *PPL* uses the perplexity filtering methods from [44], selecting pretraining data by evaluating the model's perplexity on candidate data samples. *RL_loss* replaces the regression model's fitting target of avg(k-ppl) with the loss of the RL phase, while *avg(k-entropy)* and *pass@k* substitute the target with the average entropy of k generated responses and the pass@k accuracy, respectively. *Worst* uses top-5 worst data mixtures selected by BridgeRL.

Implementation details. For small-scale proxy models, we train a total number of 512 Transformer-based decoder-only language models with 1M parameters for 1B tokens, calculating their avg(k-ppl)

scores, and then fit a regression model on basis of these scores. We then use the fitted models to calculate the Spearman correlation for the 1M, 60M and 1B parameter models on unseen data mixtures, and train a total of 15 1B-parameter models on the predicted best, worst and random data mixtures, with each model trained on 25B tokens. Following the methodology of [5], we scale the token distribution by values ranging from 0.1 to 5.0 to construct a wide spectrum of sparse and near-uniform distributions. These scaled distribution vectors are then used as the hyperparameters α for the Dirichlet sampling. For larger scales, we follow the same procedure above to perform regression fitting on 300 proxy models with 60M parameters trained on 1B tokens, and calculate the Spearman correlation coefficients with 60M and 3B parameter models on unseen data mixtures. Finally, we similarly train 15 models with 3B parameters, each trained on 25B tokens. LightGBM is adopted as the unified regression model for all modeling tasks across different model scales. We provide pre-trained and RL-evaluated small-scale proxy models (1M and 60M parameters) at this link [3]. Users can directly load these weights to extract proxy metrics ($avg(k - ppl)$) and run regression fitting for data mixture prediction.

We pre-train the decoder-only base models with 1 epoch, using the Adam optimizer in a batch size of 128K with a learning rate of 4e-4. The weight decay is 0.005. For the reinforcement learning stage, we employ both representative online (GRPO [18] following Deepseek-r1 [2]) and offline (DPO [19]) RL algorithms. Due to space constraints, the main text presents part of the experimental results, while the other experimental results are provided in the Appendix, including ablation studies for hyperparameters in Appendix D.9. We guide the learning of the model by quantifying the quality of its answers, scoring based on the output format and the final answer. k responses for calculating the avg(k-ppl) is set to 128. Our setup consists of a four-core CPU and eight NVIDIA Tesla A100 GPUs.

4.2 The performance of the regression model

We first evaluate the regression model's ability, including the fitting results over different regression fitting target models, and the performance in predicting the effect of unseen data mixtures.

Table 1: We demonstrate the correlation of LightGBM-regressed models across scales on unseen data mixtures: the model fitted with 1M parameters is used to predict the 1M, 60M and 1B parameter models, respectively, while that fitted with 60M parameters predicts the 60M and 3B parameter models. ρ denotes the correlation coefficient between the predicted and actual results of the data mixtures.

	Model Size		ρ
LightGBM	1M >	1M	86.46
		60M	69.12
		1B	58.33
	60M >	60M	77.85
		3B	68.23

(1) Fitting results over different settings. We fit the regression model using data from small-scale proxy models with 1M and 60M parameters, evaluating predictive accuracy on unseen mixing ratios at the same scales. Preliminary experiments (detailed in AppendixD.7) lead us to select LightGBM as the regressor, as it consistently outperform alternatives like Ridge Regression and Random Forest. As shown in Table 1, the model achieves a Spearman correlation (ρ) of 86.46% in the 1M-to-1M setting, while for 60M parameter models, the correlation reaches 77.85%, validating the effectiveness of our regression-based forecast.

The regression models are subsequently leveraged to evaluate the framework's extrapolation capabilities across significantly larger parameter scales and training volumes. Specifically, models fitted on 1M proxy data are used to predict performance for 60M and 1B models, while the 60M-based regressor forecasts outcomes for 3B-parameter models. In all cases, predictions are conducted on unseen data mixtures to rigorously test cross-scale transferability.

Table 1 illustrates our framework's cross-scale predictive efficacy. The 1M-proxy regression model maintains consistent Spearman correlations (ρ) of approximately 70% and 60% when extrapolated to 60M and 1B scales, respectively, while the 60M-proxy model achieves 70% correlation for the 3B

[3] `https://github.com/736619821/BridgeRL`

architecture. These results demonstrate LightGBM's robustness in capturing scaling trends across multiple orders of magnitude. We attribute minor performance fluctuations to distributional shifts and increased complexity inherent in larger training volumes (e.g., 25B tokens). Ultimately, these findings corroborate the rank-invariance-based approach for forecasting optimal data mixtures in large-scale regimes.

(2) Deep analysis of the fitted LightGBM performance on predicting the data mixture. To further analyze the distribution of the top-P mixtures, we separately visualize their weight distributions across all three domains (general, code, and math). As shown in Figure 3(b) with P=50, we observe that top-ranked data mixtures consistently contained a higher proportion of domain code up to the range of 70% to 80%, while the proportion of general text stays close to 20%. This phenomenon aligns with the findings reported in [10]'s study.



(a) P=5 (b) P=50

Figure 3: The distribution of Top-P (P=5, 50) best and worst mixtures across three domains.

4.3 The impact of predicted mixture on downstream tasks

We validate the efficacy of BridgeRL on downstream reasoning tasks. The selected benchmarks are prevalently adopted by released LLMs [21, 2] and comprehensively cover tasks across three domains: code, mathematics, and general reasoning, enabling a holistic evaluation of BridgeRL's capabilities. We compare the BridgeRL predicted data mixture with four baselines: HUMAN, PPL, Worst, and RL_loss, as introduced in experimental setups. We first utilize BridgeRL to predict the top 5 best and worst data mixture strategies. We train 1B parameter base models using the predicted data mixture and then use both GRPO and DPO to get RL-trained models separately. We further scale the model up to 3B with GRPO-trained RL models and list the results in Appendix D.3.

Table 2: Average performance of 1B models trained by GRPO, using the top-5 best data mixtures selected by BridgeRL, compared with several other data selection baselines. BSET ON indicates how many of all tasks the model achieves the best results in. The reported performance on each task is the average score of 3 runs, while the highest score on each task is highlighted in bold.

	Domain	**General**			**Code**		**Math**		**Avg.**	**Best On**
	Tasks	OBQA	PIQA	Hellaswag	Humaneval	MBPP	MATH	GSM8K		
	Metrics	Acc.	Acc.	Acc.	Pass@10	Pass@10	Acc	Acc		
Baselines	**HUMAN**	17.8	14.3	22.9	7.3	0.6	0.4	0.5	9.1	0
	PPL	**18.8**	17.2	24.6	9.2	1.4	0.6	**2.8**	10.7	2
	avg(k-entropy)	16.2	10.6	24.9	4.3	0	0.2	2.3	8.4	0
	pass@k	14.3	20.6	22.2	11.0	4.0	0.3	0.6	10.4	0
	RL_loss	5.8	**24.5**	23.0	20.1	6.0	0.9	1.9	11.7	1
BridgeRL	**Worst**	18.5	16.7	25.1	16.9	2.7	0.6	0.3	11.5	0
	Best	12.5	21.2	**25.3**	**36.8**	**10.5**	**1.1**	0.9	**15.5**	4

Table 2 lists the average performance of 1B models trained by GRPO, using the top-5 best data mixtures selected by BridgeRL. The results with DPO are listed in the Table 4 in Appendix D.2. According to Table 2, pretraining data mixtures using the proposed optimization objective avg(k-ppl) have an obvious impact on reasoning downstream tasks. Compared to the four baselines, BridgeRL demonstrates a superior performance, outperforming all four competing baselines in 4 out of 7 tasks while achieving the highest average score (3.8% higher than RL_loss). The average performance improvement across all the tasks highlights both the crucial role of data mixtures in the pretraining+RL pipeline and the validity of our data selection method. Note that the RL_loss baseline shows suboptimal performance, indicating that the RL loss function alone is not an adequate metric

for the pre-train+RL pipeline. We also notice the performance decline on GSM8K. The potential reason may be its data format deviating significantly from the training data OpenWebMath. Similar phenomena have been found in other works [10]. Besides, the settings of our RL stage comprise GSM8K training data. The performance decay may reflect prior research of the RL narrowing the reasoning bounds of the base model [20]. This also indicates the complexity of the relationship between pre-training and RL, which needs further exploration.

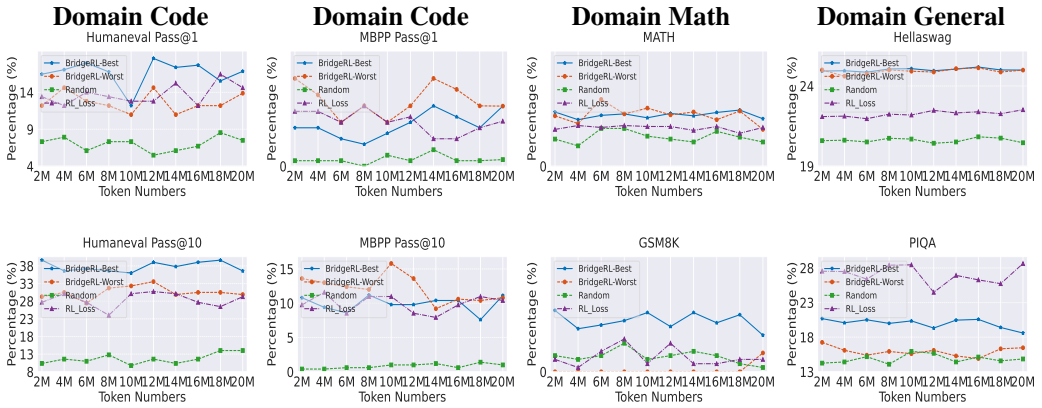


Figure 4: Three domains' downstream performances of the BridgeRL-optimal 1B parameter models (RL-trained with DPO) with top-5 best data mixtures, compared with the random sampling, RL_loss and BridgeRL selected worst baselines. The x-axis is the growing training tokens in the RL stage.

We further plot the top-5 best data mixture ratio distributions as illustrated in Figure 3(a). The best data mixture strategies have a relatively high ratio on the code domain and a similar low ratio on both general and math domains, while yielding performance improvements across all downstream tasks.

4.4 How does the performance of models with different data mixtures improve during RL?

We are interested in how the RL-trained models favor the domain distribution with the accumulated RL steps. As shown in Figure 4, we compare trajectories during RL of the models pretrained with data mixtures selected by BridgeRL (top-5 best) with the random sampling, BridgeRL-worst, and RL_loss under the DPO algorithm. For the code domain, we compare the pass@k (k=1,10) trajectories over MPBB and Humaneval. BridgeRL-best demonstrates strong RL gains from the starting point on Humaneval, eventually achieving the largest RL reasoning gains for both the pass@1 and pass@10 evaluations on all the Code tasks. For both MATH and GSM8K, BridgeRL-best demonstrates a clear advantage during RL. It can be observed that the uptrend of the RL gains for the MPBB and MATH benchmarks with the BridgeRL-optimized data mixture. For many benchmarks, we notice that the RL_loss baseline displays initial performance gains while fluctuating as the RL steps accumulate. These results further reflect the recent findings of the entropy collapse phenomenon for RL phase [45]. It also proves that the performance in the RL phase is not the sole factor affecting downstream knowledge expression and generation, finding a suitable base model also plays a significant role. The results with GRPO are listed in Figure 7 in the Appendix D.4.

4.5 Is it worth using the proxy models?

We report the GPU hours and FLOPs required to fit 512 1M-parameter models for predicting the optimal data mixtures, and compare them against other baselines. As listed in the Table 3, the computational cost of our approach remains within an acceptable and marginal range.

Our target is to predict the optimal data mixture for larger base models, thereby saving the cost of retraining the base models to get the optimal one. Notably, the training cost of the proxy model is reduced by 50% compared to the total cost of training a 1B-parameter base model on 25B tokens. This effect would become more pronounced as the training data and base model scale increase. Therefore, the choice of proxy models should be viewed as a trade-off between computational efficiency and final model performance.

Table 3: Computational consumption of BridgeRL, compared with other baselines.

Method	Stage	FLOPs	GPU hours
PPL	Pre-training	1.95E+19	4.08
RL_loss	Total	7.45E+19	16.14
Human	Pre-Training	0	0
BridgeRL	Pre-training	3.07E+18	0.64
	RL	7.14E+19	15.5
	Total	7.45E+19	16.14

4.6 Does avg(k-ppl) capture true RL-readiness beyond final scores?

To further demonstrate that $avg(k-ppl)$ captures genuine "RL-readiness" and serves as a superior regression target, we perform two complementary analyses.

First, we conduct a joint analysis of $avg(k-ppl)$, sequence-level $D_{KL}(\pi_{RL}||\pi_{base})$, and post-RL accuracy to investigate whether $avg(k-ppl)$ reflects true RL-readiness rather than simply policy similarity between the base and RL-trained models. As shown in Figure 5(a), the mixture with a lower $avg(k-ppl)$ exhibits a greater KL divergence while simultaneously achieving superior post-RL accuracy, contrary to the prediction of the proximity hypothesis.

Second, we evaluate training stability through a controlled comparison of two mixtures featuring substantially different pre-RL $avg(k-ppl)$ values but comparable initial performance on Humaneval Pass@10. As shown in Figure 5(b), the lower-$avg(k-ppl)$ mixture consistently outperforms the higher one across all checkpoints, yielding a larger overall gain (+17.28 vs. +13.41). Specifically, the higher $avg(k-ppl)$ mixture suffers from an intermediate performance drop at the 50% step, while the lower $avg(k-ppl)$ progresses steadily.

Furthermore, Figure 5(c) reveals that the lower $avg(k-ppl)$ mixture achieves a higher and significantly more stable training reward (0.78 ± 0.14 vs. 0.55 ± 0.33). These findings confirm that $avg(k-ppl)$ reliably predicts fine-grained RL training dynamics and stability rather than merely correlating with final performance.

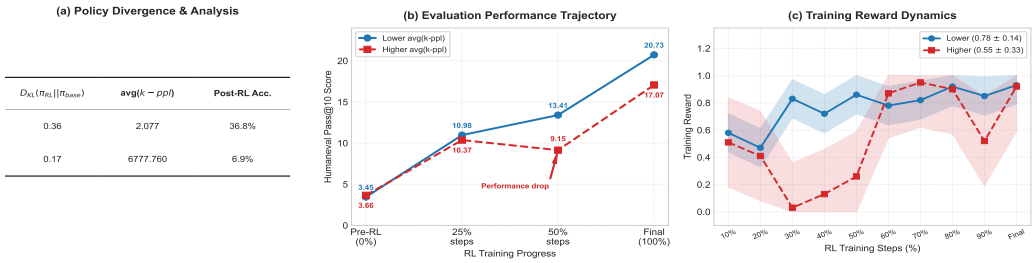


Figure 5: Joint analysis of policy divergence and training stability across mixtures with differing pre-RL $avg(k-ppl)$ values. (a) Comparison of policy divergence $D_{KL}(\pi_{RL}||\pi_{base})$, $avg(k-ppl)$, and post-RL accuracy. (b) HumanEval Pass@10 trajectories across training checkpoints. (c) Training reward dynamics (Mean $\pm$ SD) over RL steps.

5 Conclusion

On the basis of recent findings, we propose BridgeRL, a framework that predicts optimal data mixture strategies to better prepare the base model for enhanced RL reasoning. Leveraging the ratios of data from different domains as features and the proposed average perplexity of RL-generated responses as the target, BridgeRL fits a regression model to predict the optimal data mixture leading to enhanced RL capabilities. We conduct experiments to validate the performance of BrigeRL and provide the first insights into how the pre-training data mixture of the base model influences downstream RL models, potentially inspiring broader discussion in the research community.

Acknowledgements and Disclosures

We gratefully acknowledge the support from JIUTIAN Research, China Mobile, which provided essential computational power, platforms, and resource support for this research. The authors declare that they have no competing interests.

References

[1] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

[2] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[3] Xingyu Dang, Christina Baek, J Zico Kolter, and Aditi Raghunathan. Assessing diversity collapse in reasoning. In *Scaling Self-Improving Foundation Models without Human Supervision*, 2025.

[4] Darsh J Shah, Peter Rushton, Somanshu Singla, Mohit Parmar, Kurt Smith, Yash Vanjani, Ashish Vaswani, Adarsh Chaluvaraju, Andrew Hojel, Andrew Ma, et al. Rethinking reflection in pre-training. *arXiv preprint arXiv:2504.04022*, 2025.

[5] Qian Liu, Xiaosen Zheng, Niklas Muennighoff, Guangtao Zeng, Longxu Dou, Tianyu Pang, Jing Jiang, and Min Lin. Regmix: Data mixture as regression for language model pre-training. *arXiv preprint arXiv:2407.01492*, 2024.

[6] Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. *Advances in Neural Information Processing Systems*, 36:69798–69818, 2023.

[7] Xue Han, Yitong Wang, Junlan Feng, Qian Hu, and Chao Deng. Loire: Lifelong learning on incremental data via pre-trained language model growth efficiently. In *International Conference on Learning Representations*, volume 2025, pages 1024–1044, 2025.

[8] Qing Wang, Xue Han, Jiahui Wang, Lehao Xing, Qian Hu, Lianlian Zhang, Chao Deng, and Junlan Feng. Multipl-moe: Multi-programming-lingual extension of large language models through hybrid mixture-of-experts. In *EMNLP (Findings)*, pages 12817–12828, 2025.

[9] Zengzhi Wang, Fan Zhou, Xuefeng Li, and Pengfei Liu. Octothinker: Mid-training incentivizes reinforcement learning scaling. *arXiv preprint arXiv:2506.20512*, 2025.

[10] Rosie Zhao, Alexandru Meterez, Sham Kakade, Cengiz Pehlevan, Samy Jelassi, and Eran Malach. Echo chamber: Rl post-training amplifies behaviors learned in pretraining. *arXiv preprint arXiv:2504.07912*, 2025.

[11] Raffel Colin, Shazeer Noam, Roberts Adam, Lee Katherine, Narang Sharan, Matena Michael, Zhou Yanqi, Li Wei, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 2020.

[12] Bandari Abhinav, Yin Lu, Hsieh Cheng-Yu, Jaiswal Ajay Kumar, Chen Tianlong, Shen Li, Krishna Ranjay, and Shiwei Liu. Is c4 dataset optimal for pruning? an investigation of calibration data for llm pruning. *arXiv:2410.07461*, 2024.

[13] Yitong Wang, Xue Han, WenChun Gao, Qian Hu, Jiahui Wang, Ziqing Wang, Lijun Mei, and Junlan Feng. Soft orthogonal low-rank adaptation for knowledge sharing in large language model continual learning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18472–18487, 2026.

[14] Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, 2017.

[15] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

[16] Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

[17] Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models. *arXiv preprint arXiv:2505.24864*, 2025.

[18] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[19] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

[20] Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025.

[21] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407, 2024.

[22] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[23] Alon Albalak, Liangming Pan, Colin Raffel, and William Yang Wang. Efficient online data mixing for language model pre-training. *arXiv preprint arXiv:2312.02406*, 2023.

[24] Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, et al. A survey on data selection for language models. *arXiv preprint arXiv:2402.16827*, 2024.

[25] Jiasheng Ye, Peiju Liu, Tianxiang Sun, Jun Zhan, Yunhua Zhou, and Xipeng Qiu. Data mixing laws: Optimizing data mixtures by predicting language modeling performance. *arXiv preprint arXiv:2403.16952*, 2024.

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

[27] Simin Fan, Matteo Pagliardini, and Martin Jaggi. Doge: Domain reweighting with generalization estimation. *arXiv preprint arXiv:2310.15393*, 2023.

[28] Niklas Muennighoff, Alexander Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin A Raffel. Scaling data-constrained language models. *Advances in Neural Information Processing Systems*, 36:50358–50376, 2023.

[29] Thomas Minka. Estimating a dirichlet distribution, 2000.

[30] Jiasheng Ye, Peiju Liu, Tianxiang Sun, Jun Zhan, Yunhua Zhou, and Xipeng Qiu. Data mixing laws: Optimizing data mixtures by predicting language modeling performance. *arXiv preprint arXiv:2403.16952*, 2024.

[31] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

[32] Richard G Brereton and Gavin R Lloyd. Support vector machines for classification and regression. *Analyst*, 135(2):230–267, 2010.

[33] Denis Kocetkov, Raymond Li, Loubna Ben Allal, Jia Li, Chenghao Mou, Carlos Muñoz Ferrandis, Yacine Jernite, Margaret Mitchell, Sean Hughes, Thomas Wolf, Dzmitry Bahdanau, Leandro von Werra, and Harm de Vries. The stack: 3 tb of permissively licensed source code. *Preprint*, 2022.

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

[35] AI Together. Redpajama: An open source recipe to reproduce llama training dataset, 2023.

[36] FlyTech. flytech/python-codes-25k, 2024.

[37] Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, Zhengyang Tang, Benyou Wang, Daoguang Zan, Shanghaoran Quan, Ge Zhang, Lei Sha, Yichang Zhang, Xuancheng Ren, Tianyu Liu, and Baobao Chang. Omni-math: A universal olympiad level mathematic benchmark for large language models, 2024.

[38] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

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

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

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

[42] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[43] Wuhan AI Research M-A-P, University of Waterloo. Map-neo: Highly capable and transparent bilingual large language model serie. *https://map-neo.github.io/*, 2024.

[44] Zachary Ankner, Cody Blakeney, Kartik Sreenivasan, Max Marion, Matthew L Leavitt, and Mansheej Paul. Perplexed by perplexity: Perplexity-based data pruning with small reference models. *arXiv preprint arXiv:2405.20541*, 2024.

[45] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

[46] Comanici Gheorghe, Bieber Eric, Schaekermann Mike, Pasupat Ice, Sachdeva Noveen, and Dhillon Inderjit. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv:2507.06261*, 2025.

[47] Ruiqiao Bai, Xue Han, Shuo Lei, Junlan Feng, Yanyan Luo, and Chao Deng. Self-attention-based graph-of-thought for math problem solving. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 6112–6125, 2025.

[48] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[49] Yunhao Tang, Zhaohan Daniel Guo, Zeyu Zheng, Daniele Calandriello, Rémi Munos, Mark Rowland, Pierre Harvey Richemond, Michal Valko, Bernardo Ávila Pires, and Bilal Piot. Generalized preference optimization: A unified approach to offline alignment. *arXiv preprint arXiv:2402.05749*, 2024.

A Limitation

One limitation is that BridgeRL relies on domain-level ratios as the sole feature and average perplexity of RL-generated responses as the proxy target, which may overlook other critical data attributes (e.g., quality, difficulty, or temporal curriculum). Additionally, due to the high computational cost of pre-training and RL fine-tuning, our main experiments are based on single-run results without error bars, which may affect statistical reliability; the framework's generalizability to broader model scales and diverse RL domains also remains to be further validated.

B The Use of Large Language Models (LLMs)

We appreciate the assistance provided by Gemini [46] in writing aid and sentence-level polishing.

C Observation Experimental Details

C.1 C4 data-task correlation

C4 data clustering distribution. First, we randomly sample 300,000 samples from the full C4 dataset. Then, combined with the metadata from raw text samples and the semantic features obtained by converting raw text samples into high-dimensional numerical vectors via the Term Frequency-Inverse Document Frequency (TF-IDF) method, we perform clustering using the K-Means algorithm, with the Euclidean distance as the similarity metric, the maximum number of iterations set to 300, and the convergence threshold set to 1e-4. After clustering, noise samples are removed to obtain 5 imbalanced clusters: cc-news, cc-blog, cc-wiki, cc-math, cc-code (ratios: 0.682, 0.162, 0.118, 0.017, 0.021). Finally, dimensionality reduction and visualization are conducted via Principal Component Analysis (PCA).

Relevance between domain data and tasks. Following the experimental setup detailed in Section 4.1, we train 266 pre-trained models with 1M parameters and their corresponding reinforcement learning variants, and conduct comprehensive evaluations across three downstream tasks: TriviaQA, Math, and MBPP. We then compute the Spearman correlation for each data type and individual downstream task separately, from which we quantify the contribution of each data type to the performance on downstream tasks.

C.2 Comparison of evaluation metrics

We extract the pre-training loss from the 266 trained 1M-parameter pre-trained models, derive the avg(k-ppl) and downstream task pass@k scores from the reinforcement learning (RL) trained models following Section 3.3, then compute the Spearman correlation to obtain the overall correlation of each metric with performance on individual downstream tasks. We additionally perform 10-fold cross-validation to calculate the mean and standard deviation of the correlation coefficients, thereby verifying the stability of the correlation between each metric and downstream task performance.

D Supplementary Experiments

D.1 Analysis of the fitted LightGBM performance on predicting the data mixture

We further explore the performance of the best-performing regression model (LightGBM) across a broader space, where the training data is generated based on GRPO RL algorithm. Figure 6 illustrates an example of simulated new data mixtures alongside their corresponding predicted targets used to determine the best data mixture. As can be seen from the figure, when the regression model achieves its optimal target value, the proportions of code and math data are relatively low.

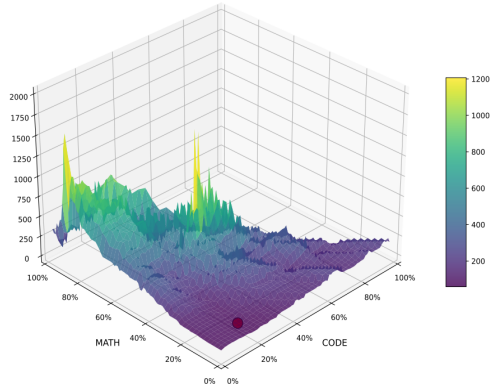


Figure 6: Simulate potential data mixtures across a broader space and use the regression model to predict the best data mixture (lowest avg(k-ppl) marked with the red dot).

D.2 The impact of predicted mixture on downstream tasks under the DPO algorithm

We further validate the efficacy of BridgeRL on downstream reasoning tasks under DPO. We compare the BridgeRL predicted data mixture with three baselines: Random, RL_loss and Worst. Random randomly selects five distinct data mixtures to train models following the pre-training + RL pipeline, and the average performance of these five models is reported as the final result. Refer to the details of other baselines in experimental setups. At first, we utilize BridgeRL to predict the top 5 best data mixture strategies. Then we train 1B parameter-based models using the predicted data mixture and use DPO to get RL-trained models separately. Table 4 lists the average score of 3 runs for the top-5 best 1B RL-trained models selected by BridgeRL.

Table 4: Average performance of 1B models trained by DPO, using the top-5 best data mixtures selected by BridgeRL, compared with several other data selection baselines. BEST ON indicates how many of all tasks the model achieves the best results in. The reported performance on each task is the average score of 3 runs, while the highest score on each task is highlighted in bold.

	Domain	**General**			**Code**		**Math**		**Avg.**	**Best On**
	Tasks	OBQA	PIQA	Hellaswag	Humaneval	MBPP	MATH	GSM8K		
	Metrics	Acc.	Acc.	Acc.	Pass@10	Pass@10	Acc	Acc		
Baselines	**Random**	7	14.8	20.5	14.0	1.0	0.5	0.1	8.3	0
	Human	8.6	6.9	22.5	20.1	2.8	0.6	**2.0**	9.1	1
	RL_loss	7.6	**28.6**	22.5	29.3	10.4	0.8	0.2	14.2	1
BridgeRL	**Worst**	**18.6**	16.5	**24.5**	29.9	10.8	0.7	0.4	14.5	2
	Best	12.5	18.6	**24.5**	**36.6**	**11.1**	**1.0**	0.7	**15.0**	**4**

Table 4 shows that pretraining data mixtures using the proposed optimization objective avg(k-ppl) have an obvious impact on reasoning downstream tasks. Compared to the three baselines, BridgeRL demonstrates a superior performance, outperforming all three competing baselines in 4 out of 7 tasks while achieving the highest average score. The average performance improvement across all the tasks highlights both the crucial role of data mixtures in the pretraining+RL pipeline and the validity of our data selection method. The RL_loss baseline also exhibits suboptimal performance under DPO, indicating that the RL loss function alone is insufficient as an evaluation metric for the pre-train+RL

pipeline. These results from both GRPO and DPO demonstrate that BridgeRL serves as a versatile framework for selecting data mixture ratios. Regardless of the reinforcement learning algorithm used, models trained with BridgeRL consistently achieve superior average performance across downstream tasks in various domains.

We also observe a performance decline on general tasks. A potential explanation is that the training data used in the RL phase exhibits a stronger bias toward in-domain content, thereby narrowing the reasoning capacity of the base model, which is a phenomenon consistent with prior research [20]. This also indicates the complexity of the relationship between pre-training and RL, which needs further exploration.

D.3 3B with GRPO-trained RL models

Table 5: Average performance of 3B models trained by GRPO, using the top-5 best data mixtures selected by BridgeRL, compared with several other data selection baselines. BEST ON indicates how many of all tasks the model achieves the best results in. The reported performance on each task is the average score of 3 runs, while the highest score on each task is highlighted in bold.

	Domain	**General**			**Code**		**Math**		**Avg.**	**Best On**
	Tasks	OBQA	PIQA	Hellaswag	Humaneval	MBPP	MATH	GSM8K		
	Metrics	Acc.	Acc.	Acc.	Pass@10	Pass@10	Acc	Acc		
BridgeRL	**Worst**	**24.6**	41.2	65.4	10.4	3.1	0.3	**1.3**	20.9	2
	Best	23.0	**44.1**	**69.0**	**44.4**	**11.8**	**0.9**	1.1	**27.8**	5

D.4 How does the performance of models with different data mixtures improve during RL under GRPO?

We are interested in how the RL-trained models favor the domain distribution with the accumulated RL steps. As shown in Figure 7, we compare trajectories during RL of the models pretrained with data mixtures selected by BridgeRL (top-5 best) with the random sampling, RL_loss and BridgeRL-worst baselines. For the code domain, we compare the pass@k (k=1,10) trajectories over MPBB and Humaneval benchmarks. BridgeRL-best demonstrates strong RL gains from the starting point, eventually achieving the largest RL reasoning gains for almost all the pass@1 and pass@10 evaluations. For the MATH benchmark, BridgeRL-best consistently maintains a significant lead throughout the training process. On the PIQA benchmark, BridgeRL and RL_Loss achieve comparable performance, with both methods consistently maintaining a leading position.

Like the results of DPO, these results also prove that the performance in the RL phase is not the sole factor affecting downstream knowledge expression and generation. Finding a suitable base model also plays a significant role.

D.5 What's a proper measurement for the data mixtures targeting RL performance?

We want to further check if the proposed avg(k-ppl) is proper for the performance measurements by evaluating its correlation with the data mixture ratios. It should be noted that there are other options, such as the downstream task accuracy and RL loss. RL loss has been proven to be inadequate in the previous analysis. We visualize the correlation between the avg(k-ppl), the potential measurement of downstream task accuracy, and the different data domains under the GRPO.

According to Figure 8(a), illustrating the correlation during pre-training stages, the avg(k-ppl) elaborates a high correlation with the data domain ratios, demonstrating its effectiveness in evaluating a suitable base model for RL. In the pre-training stage, data from the math domain provides gains to all other domains. The general domain significantly influences both code_ppl and math_ppl, which are calculated as the avg(k-ppl) from the code and math domains separately. For the downstream tasks illustrated in Figure 8(b), the data from the code source exhibits a pronounced negative impact on OBQA, which gives an explanation of the OBQA performance decline and performance of code benchmarks increase in Table 2. The visualization results highlight the importance and inherent complexity of determining the optimal pre-training data mixture and give hints for the further improvement of the BridgeRL method.

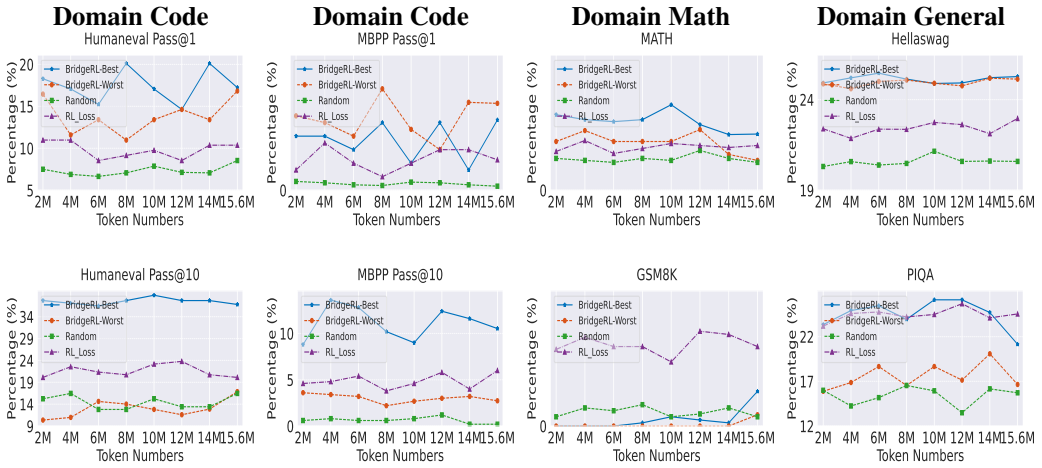


Figure 7: Three domains' downstream performances of the BridgeRL-optimal 1B parameter models with top-5 best data mixtures, compared with the random sampling method, RL_loss and BridgeRL-worst baselines under GRPO algorithm. The x-axis is the growing training tokens in the RL stage.

In addition, we visualize the correlation between the potential measurement of downstream task accuracy and the different data domains under the DPO.

For the downstream tasks illustrated in Figure 9(b), beyond general-domain benchmarks such as OBQA and PIQA, the proportion of code data exhibits a positive correlation with performance on most downstream tasks. This outcome is particularly evident on the HumanEval benchmark, which aligns well with our intuitive expectations. An increase in the proportion of math data proves beneficial to the performance on the OBQA task to some extent, demonstrating that enhanced mathematical capability can effectively improve the model's logical reasoning and structured expression abilities [47]. The visualization results highlight the importance and inherent complexity of determining the optimal pre-training data mixture and give hints for the further improvement of the BridgeRL method.

The visualization results highlight the importance and inherent complexity of determining the optimal pre-training data mixture and give hints for the future improvement of the BridgeRL method.

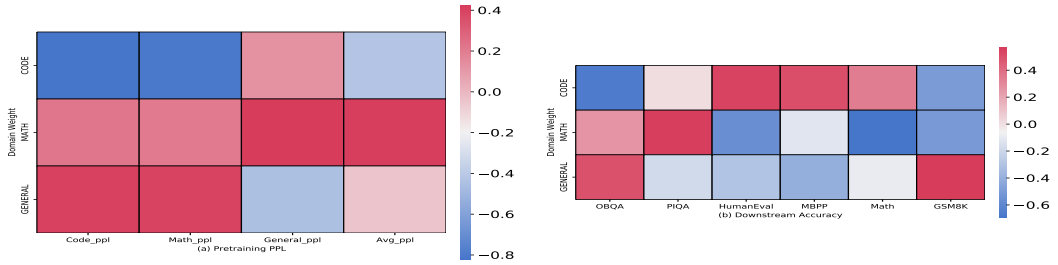


Figure 8: (a) The correlations between avg(k-ppl) and training data ratios across three domains separately and averaged over 1B models. (b) The correlations between downstream accuracy and training domain ratios on 1B models.

D.6 Why is Avg(k-ppl) superior to other fitting targets?

RL_loss: We further analyze the correlation between avg(k-ppl) and RL_loss with downstream tasks, as well as the correlation between avg(k-ppl) and RL_loss themselves, to substantiate why we choose avg(k-ppl) as the fitting target.

The higher the correlation between metrics, the larger the absolute value. As shown in Figure 10, choosing avg(k-ppl) as the target for fitting yields noticeably greater advantages and gains on downstream tasks compared to directly fitting the RL loss ($|-0.57| > |-0.37|$). Additionally,

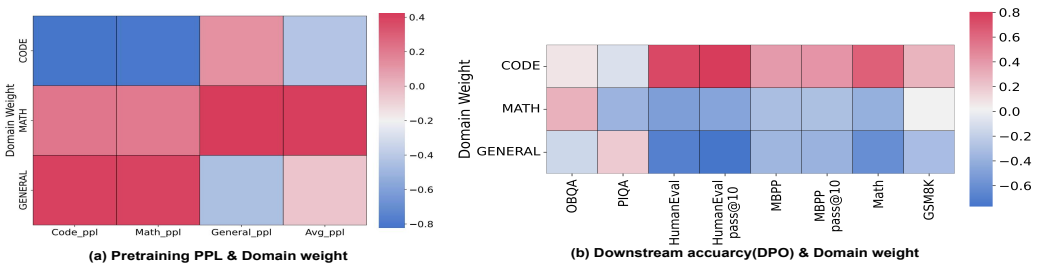


Figure 9: (a) The correlations between avg(k-ppl) and training data ratios across three domains separately and averaged over 1B models. (b) The correlations between downstream accuracy and training domain ratios on 1B models.

the positive correlation between avg-ppl and RL loss ($value = 0.35$) indicates that avg(k-ppl) and RL_loss influence the model in the same direction. These empirical factors make avg(k-ppl) our first choice for fitting the regression model.

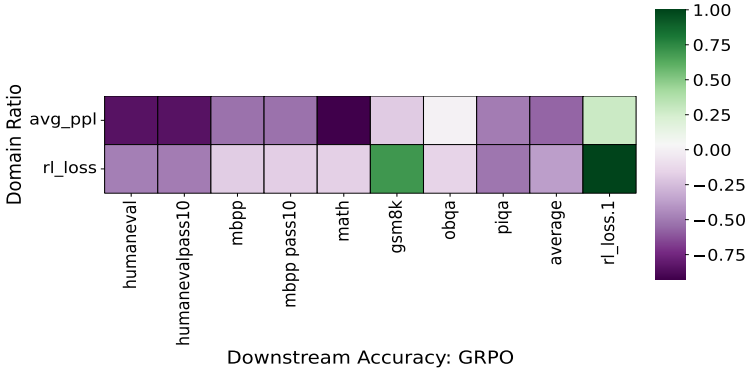


Figure 10: The correlations between avg(k-ppl) and RL_loss with downstream tasks over 1B models, under GRPO algorithm.

avg(k-entropy): We also conduct experiments with the same settings to compare the fitting targets avg(k-entropy) with avg(k-ppl). The experimental results are listed in Table 6. According to the results, avg(k-ppl) surpasses the entropy metrics in downstream tasks.

Table 6: Average performance of 1B models trained by GRPO, using the top-5 best data mixtures selected by BridgeRL, compared with avg(k-entropy). BSET ON indicates how many of all tasks the model achieves the best results in. The reported performance on each task is the average score of 3 runs, while the highest score on each task is highlighted in bold.

Domain	**General**			**Code**		**Math**			
Tasks	OBQA	PIQA	Hellaswag	Humaneval	MBPP	MATH	GSM8K	**Avg.**	**Best On**
Metrics	Acc.	Acc.	Acc.	Pass@10	Pass@10	Acc	Acc		
avg(k-entropy)	16.2	10.6	24.9	4.3	0	0.2	2.3	8.4	2
avg(k-ppl) (BridgeRL)	12.5	**21.2**	**25.3**	**36.8**	**10.5**	**1.1**	0.9	**15.5**	5

Downstream metrics: Using the downstream metrics, like the pass@k metric could also evaluate the performance. However, we argue that using downstream-task pass@k as the regression target may introduce several inherent limitations:

- High Noise and Instability: pass@k is inherently a discrete and high-variance metric, particularly on code, math, or multi-step reasoning tasks with limited evaluation samples, which injects significant noise into the regression targets and weakens the regressor's ability to learn stable performance trends.
- Strong nonlinearity with limited samples: The relationship between pass@k and the model's true capability is often highly nonlinear and even piecewise. The regression model may

struggle to capture this structure, leading to underestimation of improvements in some regions or overestimation of gains in the high-performance regime.

- Lack of Generalizable Capability: Using pass@k as the regression target effectively ties the objective to a particular task, prompt format, and evaluation configuration. As a result, the inferred optimal data mixture may overfit to idiosyncrasies of that specific downstream task, and therefore may not generalize well to other tasks or to large-scale RL training.

D.7 How do fitting results vary across different regression algorithms?

We fit the regression model using small proxy models with 1M parameters and evaluate the loss prediction performance on small models with the same parameters. As illustrated in Table 7, we adopt three representative regression algorithms: LightGBM, Ridge Regression, and Random Forest. LightGBM surpasses the other two regression methods in predicting the unseen data mixing ratio over the same 1M parameter models, achieving 86.46% in the ρ metric and 0.34 in the MSE metric.

Table 7: We separately fit three regression models based on the results of 1M parameter models trained on 1B tokens and validate them on unseen data mixtures for 1M, 60M, and 1B models. The 60M parameter model is trained with 1B tokens, while the 1B model is trained with 25B tokens. ρ compares the predicted and actual data mixtures, while MSE measures the differences between the target performances of the predicted and actual models.

Model Size	**1M**		**60M**	**1B**
Metrics	ρ	**MSE**	ρ	ρ
LightGBM	86.46	0.34	69.12	58.33
Ridge Regression	83.12	0.37	61.65	51.67
Random Forest	73.24	0.45	38.8	8.33

D.8 Can BridgeRL Outperform Human-Designed Domain-Biased Mixtures?

We conduct additional experiments comparing our predicted best mixture against various human-designed domain-biased mixtures: General-heavy, Code-heavy, and Math-heavy settings, under the same computational budget. Table 8 clearly demonstrates that algorithmic search via proxy metrics is effective.

Table 8: Performance comparison between BridgeRL and human-designed domain-biased mixtures under the same computational budget.

Mixture	**OBQA**	**PIQA**	**HumanEval**	**MBPP**	**Math**	**GSM8K**	**Avg**
General-heavy	16.35	19.82	1.83	0.60	0.08	0.06	6.46
Code-heavy	10.80	13.36	3.05	0.60	0.10	1.27	4.86
Math-heavy	8.20	10.45	1.22	0.40	0.12	1.29	3.61
BridgeRL	**12.50**	**21.20**	**36.80**	**10.50**	**1.10**	**0.90**	**13.80**

D.9 Is BridgeRL stable under different sampling settings and regression configurations?

We conducted additional experiments to evaluate the stability of our results with respect to both RL training seeds and regression configurations. As shown in Table 9, we select one fixed top-ranked mixture and one fixed bottom-ranked mixture based on BridgeRL's predictions. These results indicate that the advantage of the BridgeRL-selected top-ranked mixture over the bottom-ranked mixture is not driven by a particular RL training seed.

We further refit LightGBM using three different hyperparameter configurations and evaluate each configuration on the same held-out unseen mixtures under the 1M-to-1M setting. As illustrated in Table 10, these results indicate that the regressor's mixture-ranking ability is robust to reasonable changes in the LightGBM configuration and does not rely on a single regression setting.

Table 9: Stability of RL training across different random seeds on HumanEval Pass@10.

	HumanEval Pass@10 Performance			
Selected Mixture	**Seed = 42**	**Seed = 123**	**Seed = 2026**	**Mean ± SD**
BridgeRL-Top-ranked	36.60	32.49	40.68	**36.59 ± 3.34**
BridgeRL-Bottom-ranked	16.46	16.34	17.68	16.83 ± 0.61

Table 10: Stability of regression performance across different LightGBM hyperparameter configurations.

Setting	**n_estimators**	**learning_rate**	**max_depth**	**num_leaves**	**Spearman ρ**
Default	200	0.050	3	15	**86.46**
Configuration 2	400	0.025	2	7	85.88
Configuration 3	100	0.100	5	31	82.98

E RL algorithm details

E.1 Preliminary

We model a LLM as a policy $\pi_\theta(y|x)$ parameterized by θ, where π_θ generates a text response $y \in Y$ conditioned on a user instruction $x \in X$. Given a prompt x, the LLM π_θ will generate response y in an auto-regressive manner:

$$\pi_\theta(y|x) = \prod_t \pi_\theta(y_t|x, y_{<t}) \tag{10}$$

where y_t denotes the t-th token in the response, and $y_{<t}$ represents the sequence of tokens preceding y_t.

RLHF: To further align the SFT model π_θ with human preferences, prior studies introduce the Reinforcement Learning from Human Feedback (RLHF) framework [48], which optimizes the following objective:

$$\mathcal{L}_r(\pi_\theta) = \mathbb{E}_{x\sim p_{data}, y\sim\pi_\theta}[r(x,y) - \beta log\frac{\pi_\theta(y|x)}{\pi_{ref}(y|x)}] \tag{11}$$

where r denotes the reward function that captures human preferences, mapping a prompt–response pair to a scalar value. π_{ref} is the reference model used to regularize π_θ via Kullback–Leibler divergence, and β is a constant that controls the strength of this regularization.

E.2 DPO

A line of RL works aims to skip the reward modeling step and may be referred to as the direct preference learning approach [49]. Among them, the direct preference optimization (DPO) algorithm is the most popular one. DPO [19] is a preference-based alignment algorithm that directly optimizes the language model toward human-preferred outputs without relying on reinforcement learning. Instead of training a reward model as in RLHF, DPO reformulates preference learning as a supervised objective over pairs of responses.

Specifically, it leverages pairs of preferred and dispreferred responses and maximizes the log-likelihood of the preferred output relative to the dispreferred one under the model's distribution. Given a prompt x with two candidate responses (y^+, y^-), where y^+ is preferred over y^- according to human feedback, DPO optimizes the policy π_θ by maximizing the following objective:

$$\mathcal{L}_{DPO}(\theta) = \mathbb{E}_{(x,y^+,y^-)}[log\sigma(\beta(log\frac{\pi_\theta(y^+|x)}{\pi_{ref}(y^+|x)} - log\frac{\pi_\theta(y^-|x)}{\pi_{ref}(y^-|x)}))] \tag{12}$$

where π_θ is the policy being optimized, π_{ref} is a frozen reference model, β is a temperature parameter controlling the sharpness of preference weighting, and $\sigma(\cdot)$ is the logistic sigmoid function.

DPO allows direct optimization of the model distribution with respect to preferences, avoiding the instability of reinforcement learning while maintaining strong alignment performance.

E.3 GRPO

Group Relative Policy Optimization (GRPO) is a reinforcement learning variant proposed in [18]. It is designed for fine-tuning large language models (LLMs) in reasoning-heavy tasks, and aims to reduce computational cost and complexity while maintaining high performance. Unlike Proximal Policy Optimization (PPO), GRPO removes the need for a separate value function (critic), instead it estimates the baseline using group-based statistics.

Given a prompt x, the old policy $\pi_{\theta_{old}}$ samples a group of G candidate responses $\{y_1, y_2, ..., y_G\}$. Each response y_i is evaluated by a reward function $r(x, y_i)$, yielding scores $\{r_1, r_2, ..., r_G\}$. The group statistics are then computed as:

$$\mu = \frac{1}{G}\sum_{i=i}^{G} r_i, \sigma = \sqrt{\frac{1}{G}\sum_{i=1}^{G}(r_i - \mu)^2} \tag{13}$$

A normalized group-relative advantage is defined for each response as:

$$A_i = \frac{r_i - \mu}{\sigma} \tag{14}$$

Let the importance sampling ratio be:

$$\rho_i = \frac{\pi_\theta(y_i|x)}{\pi_{\theta_{old}}(y_i|x)} \tag{15}$$

The GRPO objective extends the PPO surrogate loss with group-relative advantages and KL regularization to a reference model π_{ref}:

$$\mathcal{L}_{GRPO}(\theta) = \mathbb{E}_{(x,y_{1,\ldots,G})\sim\pi_{\theta_{old}}}[\frac{1}{G}\sum_{i=1}^{G} min(\rho_i A_i, clip(\rho_i, 1-\varepsilon, 1+\varepsilon)A_i) - \beta D_{KL}(\pi_\theta(\cdot|x)||\pi_{ref}(\cdot|x))] \tag{16}$$

where ε is the clipping parameter that constrains policy updates, and β controls the strength of KL regularization.

GRPO eliminates the need for an explicit value function, replacing it with group-relative statistics as the baseline, thereby simplifying optimization while maintaining stability.

F Details of models and samples

F.1 Model details

The structures of the series of models are displayed in Table 11. These models utilize the same vocabulary as Qwen [22], with a vocabulary size of 151,936.

Table 11: The model structure from different LLMs.

Model	1M	60M	1B
Vocabulary Size	151936	151936	151936
n_{layers}	2	10	20
n_{heads}	8	8	16
$n_{embedding}$	256	768	2048
d_{model}	512	1536	5632

F.2 Sample analysis

In the main paper, we hypothesize that the performance drop on GSM8K may stem from its significant deviation in data format compared to the training data, OpenWebMath. To further substantiate this claim, we extract and compare sample instances from both datasets to illustrate their differences. OpenWebMath and GSM8K differ significantly in format, content, and purpose. OpenWebMath is a large-scale, web-crawled mathematical text corpus designed for pretraining, containing a mix of continuous mathematical content (e.g., LaTeX equations, proofs). In contrast, GSM8K is a curated dataset of grade-school-level math word problems, formatted as question-answer pairs with explicit step-by-step reasoning.

Table 12 shows the significant differences between OpenWebMath and GSM8K. The performance degradation on GSM8K may be attributed to the following potential factors based on our analysis:

- Pretraining on OpenWebMath may inadequately prepare models for GSM8K's task-specific format.
- OpenWebMath's focus on declarative mathematical knowledge might not transfer to GSM8K's procedural reasoning demands.

Table 12: Samples from OpenWebMath and GSM8K.

Dataset		Sample
OpenWebMath	Question:	What is the derivative of $f(x) = (e^{2x})(ln(x))$?
	Answer:	$f'(x) = e^{2x}\left(2\ln x + \frac{1}{x}\right)$ Explanation: The derivative of $\ln x$ is $\frac{1}{x}$ The derivative of $e^g(x)$is $e^g(x) \cdot g'(x)$ The derivative of $h(x) \cdot l(x)$ is $h'(x) \cdot l(x) + h(x) \cdot l'(x)$ Then $f'(x) = e^{2x} \cdot 2 \cdot \ln x + e^{2x} \cdot \frac{1}{x} = e^{2x}\left(2\ln x + \frac{1}{x}\right)$
GSM8K	Question:	Bella has two times as many marbles as frisbees. She also has 20 more frisbees than deck cards. If she buys 2/5 times more of each item, what would be the total number of the items she will have if she currently has 60 marbles? Let's think step by step.
	Answer:	When Bella buys $2/5$ times more marbles, she'll have increased the number of marbles by $2/5 * 60 = 24$. The total number of marbles she'll have is $60 + 24 = 84$. If Bella currently has 60 marbles, and she has two times as many marbles as frisbees, she has $60/2 = 30$ frisbees. If Bella buys $2/5$ times more frisbees, she'll have $2/5 * 30 = 12$ more frisbees. The total number of frisbees she'll have will increase to $30 + 12 = 42$. Bella also has 20 more frisbees than deck cards, meaning she has $30 - 20 = 10$ deck cards. If she buys $2/5$ times more deck cards, she'll have $2/5 * 10 = 4$ more deck cards. The total number of deck cards she'll have is $10 + 4 = 14$. Together, Bella will have a total of $14 + 42 + 84 = 140$ items. The answer is 140.

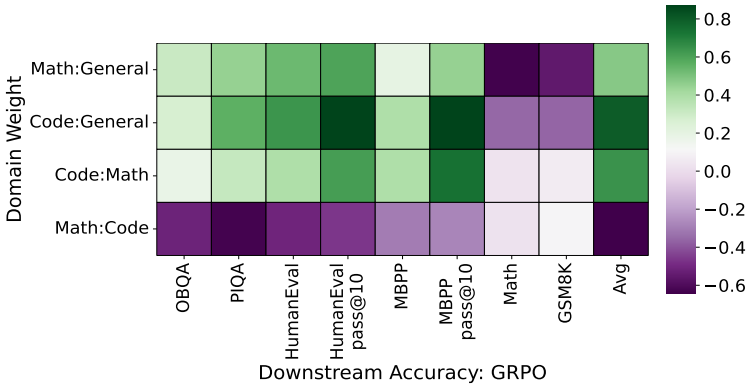


Figure 11: The relationship between the ratios of pre-training data sources (general, code, and math) and downstream tasks under GRPO.

We use a heatmap to show the relationship between the ratios of pre-training data sources (general, code, and math) and downstream tasks under GRPO. As illustrated in Figure 11, it can be clearly observed that increasing the ratios of Math to General, Code to General, and Code to Math data leads to positive effects on most tasks except for the Math-related tasks. In other words, a higher proportion of math-domain pre-training data tends to degrade performance on the GSM8K task. This observation supports our hypothesis that there exists a noticeable gap between OpenWebMath (which is more oriented toward theorems and proofs) and GSM8K&Math (which focuses more on arithmetic operations such as addition and subtraction) in terms of data format and knowledge coverage.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction explicitly state our main contributions: (1) we propose BridgeRL, a framework to find optimal data mixing ratios; (2) The results yield superior base models for RL fine-tuning. These claims are supported by our theoretical analysis (Section 3) and experiments on downstream tasks (Section 4).

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss the limitations in Appendix A, noting that BridgeRL relies solely on domain ratios and perplexity as a proxy, which may overlook other data attributes, and its generalizability to broader model scales and RL domains remains to be validated.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: We provide the full set of assumptions and complete proofs for all theoretical results in Section 3.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: We fully disclose the main implementation details, hyperparameters, data mixture strategies, and experimental protocols necessary to reproduce the main results, as described in the paper and appendix.

Guidelines:

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

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: We provide anonymized code in the supplemental material.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We fully specify all training and test details, including data splits, hyperparameters, optimizer choice, and how they were selected, in Section 4.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: Due to the high computational cost of pre-training and RL fine-tuning, our main experiments are based on single-run results without error bars; we acknowledge this limitation and discuss it in Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

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

8. **Experiments compute resources**

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

   Answer: [No]

   Justification: We provide the GPU types and counts used for our experiments.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: This research uses only publicly available datasets and involves no human subjects, sensitive personal data, or harmful applications, conforming with the NeurIPS Code of Ethics in all respects.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [No]

    Justification: This paper focuses on foundational technical contributions to data mixture optimization.

Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: This paper focuses solely on proposing a data mixture optimization framework, posing no such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We properly cite the original sources of all publicly available datasets and models used in our experiments.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

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

13. **New assets**

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

    Answer: [N/A]

    Justification: This paper does not introduce or release new datasets, code repositories, or model checkpoints; it proposes a methodological framework using existing publicly available resources.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing or research with human subjects; all experiments are conducted using publicly available datasets and automated model training pipelines.

    Guidelines:

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

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

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

    Answer: [N/A]

Justification: All experiments are conducted using publicly available datasets and automated model training pipelines.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: LLMs were used solely for writing assistance and language polishing, as disclosed in Appendix B.

Guidelines:

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