PSPO: Prompt-Level Prioritization and Experience-Weighted Smoothing for Efficient Policy Optimization

Xinxin Zhu[1, 2], Ying He[1*], Haowen Hou[2], Ruichong Zhang[3], Nianbo Zeng[1, 2], Yulin Peng[1], Jiongfeng Fang[2], F. Richard Yu[4]

[1]College of Computer Science and Software Engineering, Shenzhen University, China
[2]Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen, China
[3]Tsinghua University, China
[4]School of Information Technology, Carleton University, Canada
2300541002@email.szu.edu.cn, heying@szu.edu.cn

Abstract

Reinforcement Fine-tuning (RFT) methods such as Group Relative Policy Optimization (GRPO) have demonstrated strong capabilities in aligning Large Language Models with human preferences. However, these approaches often suffer from limited data efficiency, necessitating extensive on-policy rollouts to maintain competitive performance. We propose PSPO (Prompt-Level **P**rioritization and Experience-Weighted **S**moothing for Efficient **P**olicy **O**ptimization), a lightweight yet effective enhancement to GRPO that improves training stability and sample efficiency through two complementary techniques. First, we introduce an experience-weighted reward smoothing mechanism, which uses exponential moving averages to track group-level reward statistics for each prompt. This enables more stable advantage estimation across training steps without storing entire trajectories, allowing the model to capture historical reward trends in a lightweight and memory-efficient manner. Second, we adopt a prompt-level prioritized sampling strategy, which is an online data selection method inspired by prioritized experience replay. It dynamically emphasizes higher-impact prompts based on their relative advantages, thereby improving data efficiency. Experiments on multiple mathematical reasoning benchmarks and models show that PSPO achieves comparable or better accuracy than GRPO, while significantly accelerating convergence, and maintaining low computational and memory overhead.

Code — https://github.com/sunCityDaa/PSPO

Introduction

Reinforcement Learning (RL) is a key technique for aligning large language models (LLMs) with human preferences, helping to mitigate biases and inaccuracies (Ouyang et al. 2022; Kumar et al. 2025; Hu 2025; Guo et al. 2025). Among RL methods, Proximal Policy Optimization (PPO) (Schulman et al. 2017) has been widely adopted due to its robustness and stability in on-policy updates (Vemprala et al. 2024). However, the reliance of PPO on value estimation requires an value model, which substantially increases training cost. (Xu et al. 2024; Lanchantin et al. 2025).

To avoid the overhead of the value model, recent REINFORCE-based methods (Williams 1992) such as GRPO (Shao et al. 2024), DPO (Rafailov et al. 2023), RLOO (Ahmadian et al. 2024), and REINFORCE++ (Hu 2025), construct surrogate objectives directly from scalar rewards or preferences (Swamy et al. 2025; Schulman et al. 2016). Among them, GRPO has shown strong performance in LLM alignment (Li et al. 2025b), by estimating advantages from multiple on-policy responses per prompt. However, this design incurs high computational costs and low data efficiency. Moreover, GRPO uniformly samples prompts during training, overlooking that samples vary in learning utility (Sun et al. 2025).

To improve the stability and sample efficiency of GRPO without significant overhead, we propose Prompt-level Sampling and Prioritization Optimization (PSPO). PSPO retains GRPO's core design but introduces two lightweight enhancements: (1) Experience-weighted Reward Smoothing (ERS), which stabilizes group-wise advantage estimates by maintaining an Exponentially Moving Average (EMA) of reward statistics per prompt; and (2) Prompt-level Prioritization Sample (PPS), which performs online data selection by dynamically adjusting sampling probabilities based on each prompt's learning utility, inspired by Prioritized Experience Replay (PER) (Schaul et al. 2015). This allows PSPO to focus on more informative prompts as training progresses (Albalak et al. 2024), leading to more stable updates, faster convergence, and better generalization, all while keeping computational and memory costs low (Di-Castro, Mannor, and Di Castro 2022).

To validate PSPO, we conduct evaluations on various math reasoning benchmarks and models, including DeepSeek-R1-Distill-Qwen-1.5B, Qwen2.5-Math-1.5B, Qwen2.5-Math-1.5B-Instruct, Qwen2.5-Math-7B, and Qwen2.5-Math-7B-Instruct, based on both GRPO and its simplified variant Dr.GRPO (Liu et al. 2025). The results show that both ERS and PPS contribute positively across settings. For example, PPS improves GRPO by up to 5.10%

*Corresponding author

on DeepSeek-R1-Distill-Qwen-1.5B, while ERS.PPS boosts Dr.GRPO by 4.08% on Qwen2.5-Math-1.5B. ERS alone brings gains of 2.70% on both Qwen2.5-Math-1.5B-Instruct (Dr.GRPO) and Qwen2.5-Math-7B (GRPO), and the ERS.PPS combination achieves 3.20% on Qwen2.5-Math-7B-Instruct (GRPO). Ablation studies further analyze the role of the smoothing factor in ERS and the prioritization strategy in PPS, confirming their effectiveness in improving optimization stability and sample efficiency.

Related Work

Data efficiency In RL with LLMs, improving data efficiency is critical due to the high cost of rollout generation. A common solution is to use a replay buffer that mixes on- and off-policy updates to reuse past trajectories (Li et al. 2025b; Sun et al. 2025; Zheng et al. 2025; Wang et al. 2025; Lu et al. 2025), though this increases storage demands and may introduce bias in gradient estimates (Lanchantin et al. 2025; Gu et al. 2017). Another direction focuses on online data selection, which prioritizes informative or difficult prompts during training (Albalak et al. 2024; Wang et al. 2024; Yu, Das, and Xiong 2024; Liao et al. 2025; Yu et al. 2025). However, recent studies (Sun et al. 2025; Kaddour et al. 2023) show these methods remain computationally expensive and scale poorly to large language models.

Exponential Smoothing in Reinforcement Learning Exponential recency-weighted averages are widely used in RL to track evolving statistics with constant per-step cost (Sutton, Barto et al. 1998). In multi-armed bandits, incremental update rules allow models to incorporate new rewards without storing complete histories. Recent methods like PTR-PPO (Liang et al. 2021) maintain trajectory-level statistics—using running estimates of mean and variance of cumulative returns to compute sampling priorities. We extend this idea to prompt-level reward statistics in RFT.

Preliminary

Group Relative Policy Optimization

GRPO operates by normalizing reward scores among responses generated for the same prompt. For a given input $q \in \mathcal{D}$, the model generates a group of outputs $\{o_i\}_{i=1}^{G}$. The advantage for each output is then computed by normalizing the group-level rewards $\mathbf{r} = \{r_1, \cdots, r_G\}$.

$$A_i = \frac{r_i - \mu}{\sigma + \epsilon} \tag{1}$$

where $\mu = \text{mean}(\mathbf{r})$, $\sigma = \text{std}(\mathbf{r})$. σ is the standard deviation of $\mathbf{r}$, ϵ is a small constant to ensure numerical stability. GRPO objective can be formulated as:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\,\{o_i\}_{i=1}^{G}\sim\pi_{\text{old}}(O|q)} \left[\frac{1}{G}\sum_{i=1}^{G}\left(\min\left(\frac{\pi_\theta(o_i|q)}{\pi_{\text{old}}(o_i|q)}A_i,\right.\right.\right. \left.\left.\left.\text{clip}\left(\frac{\pi_\theta(o_i|q)}{\pi_{\text{old}}(o_i|q)}, 1-\epsilon, 1+\epsilon\right)A_i\right) - \eta\,\mathbb{D}_{\text{KL}}(\pi_\theta\|\pi_{\text{ref}})\right)\right] \tag{2}$$

Algorithm 1: PSPO algorithmn

Input: Initial policy π_{init}; prompt set $\mathcal{D}$; hyperparameters ϵ, η, EMA coefficient α, PER coefficients λ, β

Initialize: Policy $\pi_\theta \leftarrow \pi_{\text{init}}$

for epoch $= 1$ to I **do**
  Set reference model: $\pi_{\text{ref}} \leftarrow \pi_\theta$
  for step $= 1$ to M **do**
    Sample batch $\mathcal{D}_b$ from $\mathcal{D}$ using priority distribution
    Set old policy: $\pi_{\theta_{\text{old}}} \leftarrow \pi_\theta$
    for each $q \in \mathcal{D}_b$ **do**
      Sample G outputs $\{o_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot \mid q)$
      Update smoothed reward stats $\bar{\mu}$, $\bar{\sigma}$ via EMA
      Compute advantages $\{\bar{A}_i\}_{i=1}^{G}$
      Update priority score for q
    end for
    Update policy π_θ by maximizing $\mathcal{J}_{\text{PSPO}}(\theta)$
  end for
end for

Here, η regulates the KL divergence between the current policy and reference policy.

Methodology

Our proposed method is two-fold: (1) **Experience-weighted reward smoothing**, which stabilizes group-wise advantage estimation by maintaining running statistics of reward mean and variance for each prompt. Unlike replay-based off-policy methods that require storing full token-level generations for reuse, ERS is lightweight—it tracks only scalar statistics—and fully preserves the on-policy nature of GRPO. This design avoids distributional shift, gradient bias, and excessive memory cost associated with full trajectory storage; and (2) **Prompt-level prioritized sample**, which improves sample efficiency by biasing prompt selection toward high-priority examples through an online data selection mechanism. Importantly, we only store the prompt identifier and its corresponding sampling priority, without retaining the full generated outputs. Full pseudocode is provided in Algorithm 1. Figure 1 shows an overview of PSPO.

Experience-weighted reward smoothing To reduce the variance in group-wise advantage estimation across training steps, we replace the instantaneous statistics (μ, σ) for each input q with their exponentially smoothed estimates $(\bar{\mu}, \bar{\sigma})$. Specifically, after prompt q has been selected t times during training, we update the smoothed mean using:

$$\bar{\mu}^t \leftarrow \bar{\mu}^{t-1} + \alpha \cdot (\mu^t - \bar{\mu}^{t-1}) \tag{3}$$

where $\alpha \in (0, 1)$ is a smoothing coefficient, and μ^t and σ^t are the current group-level mean and standard deviation.

The smoothed standard deviation is updated using the following exponential moving formula (Finch 2009):

$$\begin{aligned}(\bar{\sigma}^t)^2 \leftarrow &(1-\alpha)(\bar{\sigma}^{t-1})^2 + \alpha(\sigma^t)^2 \\ &+ \alpha(1-\alpha)(\mu^t - \bar{\mu}^{t-1})^2\end{aligned} \tag{4}$$

The first two terms blend the previous and current variances, while the third term corrects for the change in the mean,

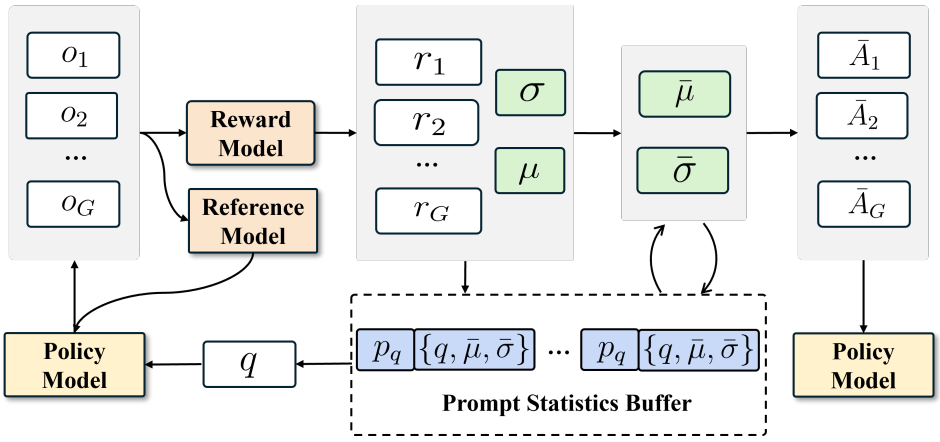

Figure 1: Demonstration of PSPO. At each training step, we compute the empirical mean μ and standard deviation σ from the group of rewards to update prompt-level priorities for online data selection. Exponential smoothing is then applied to obtain $\bar{\mu}$ and $\bar{\sigma}$, which are used to normalize advantages for stable policy updates. The statistics buffer stores $\bar{\mu}$ and $\bar{\sigma}$ to ensure consistent normalization when reusing past samples, along with priority scores to enable prioritized sampling in future updates.

ensuring that the variance estimate remains accurate even as the underlying reward distribution shifts. This adjustment makes the estimator more adaptive and robust.

This smoothing scheme can be viewed as implicitly maintaining a prompt-specific value estimate V^t, corresponding to the EMA-smoothed mean reward $\bar{\mu}^t$ for prompt q after it has been selected t times. Inspired by incremental estimation techniques for nonstationary environments in RL (Sutton, Barto et al. 1998), this value is updated online as:

$$V^t = V^{t-1} + \alpha \left[\mu^t - V^{t-1}\right] \tag{5}$$

This update rule forms an exponential recency-weighted average (Sutton, Barto et al. 1998), enabling the value estimate to adapt to reward shifts over time. The smoothed mean and variance estimates $(\bar{\mu}^t, \bar{\sigma}^t)$ are then used to compute a more stable advantage:

$$\bar{A}_i = \frac{r_i - V^t}{\bar{\sigma}^{t-1} + \epsilon} \tag{6}$$

This effectively casts PSPO as a REINFORCE with Baseline algorithm, where the EMA-smoothed mean reward V^t serves as a learned, prompt-specific baseline. This approach reduces the variance of gradient estimates without introducing bias, thereby improving training stability.

Prompt-level prioritized sample We apply prompt-level prioritization to improve sample efficiency via online data selection, focusing on high-impact prompts. In our setting, each prompt is associated with a group of sampled outputs and their corresponding reward scores. Inspired by PER in value-based RL—where transitions are ranked by temporal-difference error—we instead compute prompt importance using advantage-based signals derived from group-level rollouts (Liang et al. 2021).

To capture different aspects of a prompt's learning potential, we explore three alternative definitions of priority:

- **Average Group-Wise Advantage**: This criterion computes the priority as the mean advantage across all sampled responses $\{A_i\}_{i=1}^G$ for prompt q, offering a stable estimate of the overall utility of the prompt:

$$p_q = \frac{1}{G} \sum_{i=1}^{G} |A_i| \tag{7}$$

- **Maximum Group-Wise Advantage**: To focus on the most promising response in each group, this variant assigns priority based on the highest observed advantage within the group:

$$p_q = \max_{i=1,\ldots,G} |A_i| \tag{8}$$

- **Last-Sample Advantage**: Motivated by online update efficiency, this approach uses only the most recently sampled output's advantage to represent the prompt's learning potential:

$$p_q = |A_G| + \epsilon \tag{9}$$

where A_G is the advantage of the last generated response in the group. ϵ is a small positive constant that prevents the edge-case of prompt not being revisited once their error is zero.

Then the probability of prioritized sampling prompt q as:

$$P(q) = \frac{p_q^\lambda}{\sum_{k \in \mathcal{D}} p_k^\lambda} \tag{10}$$

The exponent λ determines how much prioritization is used, with $\lambda = 0$ corresponding to the uniform case (Schaul et al. 2015).

However, prioritized sampling introduces a known distributional bias, since the training distribution deviates from the original on-policy data. This bias arises because samples with higher advantages are overrepresented, distorting the empirical expectation of the policy gradient. To correct for this, we apply importance sampling weights, defined as the

Policy Model	AIME24	MATH500	AMC23	Minerva	OlympiadBench	Overall Avg
DeepSeek-R1-Distill-Qwen-1.5B	30.0	81.4	67.5	**30.9**	51.9	52.34
GRPO	23.3	83.0	75.0	28.3	**54.1**	52.74
+ ERS	23.3	84.8	72.5	30.5	52.4	52.70
+ PPS	**40.0**	**85.0**	**80.0**	30.5	53.9	**57.88**
+ ERS.PPS	23.3	**85.0**	65.0	29.4	53.5	51.24
Qwen2.5-Math-1.5B	0.0	40.8	**37.5**	9.9	**23.1**	22.26
GRPO	**6.7**	**44.2**	35.0	7.4	21.8	**23.02**
+ ERS	**6.7**	43.2	30.0	12.9	21.8	22.92
+ PPS	3.3	42.6	27.5	9.2	22.2	20.96
+ ERS.PPS	3.3	39.8	25.0	**13.2**	22.2	20.70
Qwen2.5-Math-1.5B-Instruct	**16.7**	73.6	42.5	**27.6**	37.0	39.48
GRPO	13.3	**73.8**	52.5	26.8	**37.3**	40.74
+ ERS	10.0	72.2	**60.0**	26.5	37.0	**41.14**
+ PPS	13.3	73.6	55.0	24.6	36.4	40.58
+ ERS.PPS	6.7	73.2	**60.0**	26.5	37.2	40.72
Qwen2.5-Math-7B	**16.7**	**60.4**	45.0	12.1	22.2	**31.28**
GRPO	3.3	56.4	**50.0**	11.8	24.1	29.12
+ ERS	10.0	58.4	35.0	13.2	23.0	27.92
+ PPS	6.7	55.0	37.5	11.8	23.5	26.90
+ ERS.PPS	10.0	58.4	45.0	**14.7**	**24.4**	30.50
Qwen2.5-Math-7B-Instruct	**13.3**	**83.2**	55.0	32.7	42.5	45.34
GRPO	6.7	81.4	47.5	**33.5**	40.1	41.84
+ ERS	10.0	81.2	**62.5**	**33.5**	**43.0**	**46.04**
+ PPS	6.7	81.4	60.0	32.0	39.3	43.88
+ ERS.PPS	10.0	81.4	55.0	32.0	40.7	43.82

Table 1: Evaluation results on five math benchmarks across different base models and training strategies. We evaluate GRPO with two extensions: ERS and PPS. ERS.PPS refers to the simultaneous application of both methods.

inverse of the sampling probability (normalized over the $\mathcal{D}$), raised to a scaling factor $\beta \in [0, 1]$.

$$w_q = \left(\frac{1}{N \cdot P(q)} \right)^{\beta} \tag{11}$$

where N is the total number of prompts in the $\mathcal{D}$. These weights downscale the gradient contributions of frequently sampled (high-priority) prompts and amplify underrepresented ones, partially restoring the unbiased gradient estimation. For stability reasons, we always normalize weights by $1/max_q w_q$ so that they only scale the update downwards (Schaul et al. 2015).

By incorporating importance sampling and the smoothed advantage $\bar{A}_i$, the PSPO objective is defined as:

$$\mathcal{J}_{\text{PSPO}}(\theta) = \mathbb{E}_{q \sim P(q),\, \{o_i\}_{i=1}^{G} \sim \pi_{\text{old}}(O|q)} \left[\frac{w_q}{G} \sum_{i=1}^{G} \min \left(\frac{\pi_\theta(o_i|q)}{\pi_{\text{old}}(o_i|q)} \bar{A}_i, \right. \right. \\ \left. \left. \text{clip} \left(\frac{\pi_\theta(o_i|q)}{\pi_{\text{old}}(o_i|q)}, 1-\epsilon, 1+\epsilon \right) \bar{A}_i \right) - \eta \, \mathbb{D}_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}) \right] \tag{12}$$

This formulation avoids the need for a separate value network while still reducing gradient variance via prompt-specific baselines.

Intropy Analysis of PSPO

The **Intropy framework** quantifies intelligence evolution as $d\mathcal{L} = \delta S / R$, where $d\mathcal{L}$ (Intropy) denotes the incremental intelligence gain, δS represents newly absorbed information, and R reflects the system's internal state, such as complexity or uncertainty (Ren et al. 2025).

Under this perspective, **PSPO** can be interpreted as an implicit entropy normalization mechanism that regulates gradient magnitude according to local uncertainty, thereby maximizing information efficiency. The Experience-Weighted Reward Smoothing (ERS) component reduces stochastic reward fluctuations, effectively lowering R by dissipating transient entropy and stabilizing learning signals. Conversely, the Prompt-Level Prioritized Sampling (PPS) component amplifies high-information prompts, increasing δS by directing computation toward states with greater informational gain.

Together, ERS and PPS cooperatively enhance the system's Intropy dynamics: ERS minimizes entropy dissipation ($R \downarrow$) while PPS increases information absorption ($\delta S \uparrow$), leading to higher incremental intelligence $d\mathcal{L}$ and more efficient policy optimization.

Policy Model	AIME24	MATH500	AMC23	Minerva	OlympiadBench	Overall Avg
Qwen2.5-Math-1.5B	0.0	40.8	37.5	9.9	23.1	22.26
Dr.GRPO	3.3	39.6	32.5	9.9	21.3	21.32
+ ERS	**6.7**	43.0	37.5	10.3	**23.4**	24.18
+ PPS	3.3	43.4	22.5	10.3	21.8	20.26
+ ERS.PPS	3.3	**44.6**	**45.0**	**12.9**	21.2	**25.40**
Qwen2.5-Math-1.5B-Instruct	**16.7**	73.6	42.5	**27.6**	37.0	39.48
Dr.GRPO	10.0	73.6	45.0	26.1	36.1	38.16
+ ERS	10.0	72.8	50.0	27.2	37.2	39.44
+ PPS	6.7	73.0	**55.0**	25.7	**39.0**	**39.88**
+ ERS.PPS	10.0	**73.8**	45.0	26.1	36.3	38.24
Qwen2.5-Math-7B	16.7	60.4	45.0	12.1	22.2	31.28
Dr.GRPO	10.0	59.2	40.0	**17.3**	21.3	29.56
+ ERS	13.3	57.0	42.5	15.1	23.4	30.26
+ PPS	6.7	**61.4**	**60.0**	11.0	20.3	**31.88**
+ ERS.PPS	**26.7**	57.8	27.5	14.7	**24.1**	30.16
Qwen2.5-Math-7B-Instruct	**13.3**	**83.2**	55.0	**32.7**	**42.5**	**45.34**
Dr.GRPO	6.7	80.8	**62.5**	31.6	39.0	44.12
+ ERS	6.7	81.8	55.0	32.4	40.0	43.18
+ PPS	10.0	80.6	57.5	31.6	40.7	44.08
+ ERS.PPS	10.0	81.2	57.5	32.4	40.3	44.28

Table 2: Evaluation results on five math benchmarks across different base models and training strategies. We evaluate Dr.GRPO with two extensions: ERS and PPS. ERS.PPS refers to the simultaneous application of both methods.

Experiment

Experiment Settings

Backbone Models and Training Datasets We conduct experiments using the Qwen series LLMs, which have shown strong performance in complex reasoning tasks (Li et al. 2025b). Following DeepSeek-R1-Zero, which demonstrates that RL can enhance LLM reasoning without SFT, we also use base models. We run our experiments on DeepSeek-R1-Distill-Qwen-1.5B (Guo et al. 2025), Qwen2.5-Math-1.5B, Qwen2.5-Math-1.5B-Instruct, Qwen2.5-Math-7B, and Qwen2.5-Math-7B-Instruct (Yang et al. 2024). Our training data are sourced from (Dang and Ngo 2025). This dataset consists of only 7000 samples with mixed problem difficulties. For Qwen2.5-Math-1.5B/7B models, we use 4096 as the context length, as it is the maximum context length for those two models. For DeepSeek-R1-Distill-Qwen-1.5B, we set the context length to 8196.

RL Training Configuration We conduct experiments on GRPO and DrGRPO using the OpenR1 (Hugging Face 2025) and TRL (von Werra et al. 2020) framework. To build strong baselines, we adopt the Clip-Higher technique from (Yu et al. 2025). Given limited computational resources (2× NVIDIA A100 40GB GPUs), we adopt DeepSpeed ZeRO Stage 2 (Aminabadi et al. 2022; Rajbhandari et al. 2020) for memory optimization. Under this setup, we apply LoRA (Hu et al. 2021) to fine-tune all linear layers of the 7B model, while full fine-tuning is feasible for the smaller 1.5B model. See the appendix for hyperparameter details. We employ a hybrid reward strategy that combines cosine reward (Dang and Ngo 2025), which scales correctness by output length using a cosine schedule to favor concise correct completions, and format reward (Guo et al. 2025).

Evaluation Benchmarks and Metrics We evaluate on five popular mathematical reasoning benchmarks, including AIME 2024, AMC 2023 (MAA: American Mathematics Competitions 2023), MATH500 (Hendrycks et al. 2021), Minerva (Lewkowycz et al. 2022) and OlympiadBench (He et al. 2024), using a rollout temperature of 0.6 and top-p sampling with p = 0.95. We adopt pass@1 as the evaluation metric to measure the performance. Note that since we employ the Qwen2.5-Math base models, which have a context length of 4K, we limit the generation budget to 3K for all compared baselines. For DeepSeek-R1-Distill-Qwen-1.5B, whose context length is 128K, the generation budget is set to 32K for all baselines.

Results

We evaluate two components: PPS using the Last-sample advantage without smoothing, and ERS with $\alpha = 0.99$. Experiments compare each component individually and in combination to assess their contributions to performance and stability based on GRPO and Dr.GRPO. The Dr.GRPO we evaluate is a minimal variant of GRPO, differing only by removing the response-level length bias, all other components remain identical.

Table 1 shows that GRPO consistently improves performance over base models, especially on Qwen2.5-Math-1.5B. ERS brings further gains, particularly for the 1.5B-Instruct and 7B-Instruct models, while PPS notably

Policy Model	Method	AIME24	MATH500	AMC23	Minerva	OlympiadBench	Overall Avg
DeepSeek-R1-Distill-Qwen-1.5B	ERS	33.3	86.2	77.5	33.5	53.6	**56.82**
	ERS.PPS	50.0	84.6	62.5	27.6	51.9	55.32
Qwen2.5-Math-1.5B	ERS	3.3	41.0	20.0	12.5	24.6	20.28
	ERS.PPS	6.7	43.8	32.5	11.4	24.7	**23.82**
Qwen2.5-Math-1.5B-Instruct	ERS	13.3	73.6	57.5	27.6	36.4	**41.68**
	ERS.PPS	13.3	73.4	47.5	24.1	37.8	39.22
Qwen2.5-Math-7B	ERS	16.7	56.4	40.0	18.6	25.0	**31.34**
	ERS.PPS	10.0	56.8	50.0	15.1	23.4	31.06
Qwen2.5-Math-7B-Instruct	ERS	10.0	82.2	52.5	32.7	38.4	43.16
	ERS.PPS	6.7	83.6	57.5	30.5	41.9	**44.04**

Table 3: Ablation study of the linear variance update in GRPO, where the smoothed variance $\bar{\sigma}^t$ is updated via a linear rule (see Equation (13)).

Policy Model	Method	AIME24	MATH500	AMC23	Minerva	OlympiadBench	Overall Avg
Qwen2.5-Math-1.5B	ERS	3.3	38.6	30.0	10.3	22.2	20.88
	PPS	3.3	41.0	27.5	9.2	23.7	20.94
	ERS.PPS	0.0	46.0	40.0	12.1	24.4	**24.50**
Qwen2.5-Math-1.5B-Instruct	ERS	13.3	73.4	55.0	23.9	38.7	**40.86**
	PPS	10.0	73.2	47.5	26.1	35.5	38.46
	ERS.PPS	10.0	73.4	52.5	24.6	39.4	39.98
Qwen2.5-Math-7B	ERS	13.3	56.2	37.5	12.5	25.5	29.00
	PPS	3.3	56.8	50.0	14.3	25.5	**29.98**
	ERS.PPS	6.7	58.0	40.0	15.4	25.5	29.12
Qwen2.5-Math-7B-Instruct	ERS	10.0	82.2	60.0	33.1	40.0	45.06
	PPS	10.0	81.4	55.0	30.1	39.7	43.24
	ERS.PPS	10.0	81.8	62.5	31.6	39.7	**45.12**

Table 4: Ablation of variance-based advantage smoothing in Dr.GRPO. We replace the adaptive variance update with $\bar{A}_i = r_i - V^t$ to isolate the effect of variance normalization.

boosts DeepSeek-R1-Distill-Qwen-1.5B. However, combining ERS and PPS doesn't always lead to additive gains, suggesting possible overlap. Overall, ERS and PPS are effective on their own. However, RFT does not always improve performance—especially for larger models like Qwen2.5-Math-7B model.

Table 2 shows that ERS, PPS, and their combination consistently outperform the GRPO baseline across Qwen2.5 base model. The ERS+PPS combo achieves the best result on the 1.5B base model, while PPS alone performs well on the 7B base and 1.5B instruct models. For the 7B instruct model, RFT shows limited gains, likely due to LoRA and small batch size. Overall, both ERS and PPS prove effective, with their combination offering further improvements.

Impact of Variance

To simplify the update and improve robustness, we switch to a more direct exponential moving average of the standard deviation:

$$(\bar{\sigma}^t)^2 \leftarrow (\bar{\sigma}^{t-1})^2 + \alpha \cdot ((\sigma^t)^2 - (\bar{\sigma}^{t-1})^2) \qquad (13)$$

We compare the linear variance update (Table 3) with exponential smoothing (Table 1). While linear update performs notably better for DeepSeek-R1-Distill-Qwen-1.5B overall, it also yields the best results for Qwen2.5-Math-1.5B with ERS+PPS, Qwen2.5-Math-1.5B-Instruct with ERS, and Qwen2.5-Math-7B with ERS.

Table 4 reports the results of Dr.GRPO without variance-based advantage smoothing, using $\bar{A}_i = r_i - V^t$ as a simplified baseline. Compared to the STD-based results (Table 2), we observe that base models generally perform worse without STD-based smoothing, indicating its importance for stabilizing training in these settings. In contrast, instruct models perform comparably or even slightly better without it, suggesting that they may benefit less from variance-based smoothing due to their stronger alignment or instruction tuning.

Impact of Prompt-Level Prioritization

We evaluate three prompt-level prioritization strategies in GRPO using DeepSeek-R1-Distill-Qwen-1.5B: (1) Average group-wise advantage, (2) Max group-wise advantage,

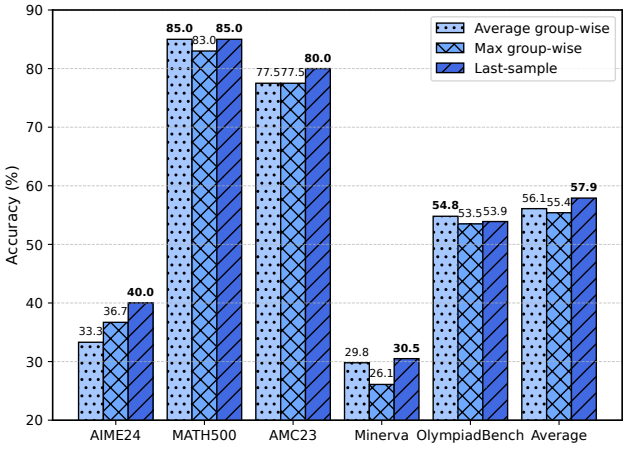

Figure 2: Effect of prompt-level prioritization strategies

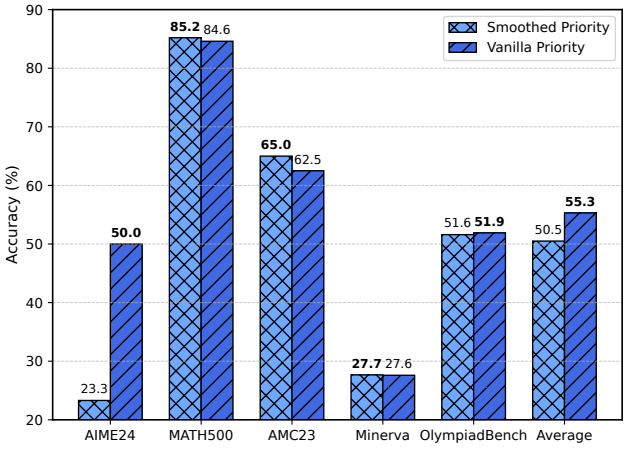

Figure 3: Effect of reward smoothing on prioritization

and (3) Last-sample advantage. As shown in Figure 2, the last-sample strategy achieves the highest average accuracy (57.9%) and performs notably better on AIME24 (40.0% vs. 33.3%), suggesting that using the most recent learning signal is especially effective for harder tasks.

We further compare two variants of Last-sample advantage: Vanilla Priority, which uses raw reward statistics (μ, σ), and Smoothed Priority, which applies EMA ($\bar{\mu}$, $\bar{\sigma}$). As shown in Figure 3, Vanilla Priority slightly outperforms Smoothed Priority on average, with a larger margin on AIME24. This suggests that tracking immediate feedback better adapts to evolving prompt difficulty during training.

Impact of Experience-weighted Reward Smoothing Factor

This coefficient controls how much historical reward statistics (mean and standard deviation) influence current advantage normalization. We denote the step size by α or $\alpha_n(q)$, n denotes the number of times sample q has been drawn. To evaluate the effect of the EMA smoothing factor, we vary $\alpha_n(q) \in \frac{1}{n}, 0.1, 0.3, 0.5, 0.7, 0.9, 0.99, 1.0$. When $\alpha_n(q) =$ 1.0, the model reduces to vanilla GRPO with no smoothing; smaller values assign more weight to past statistics.

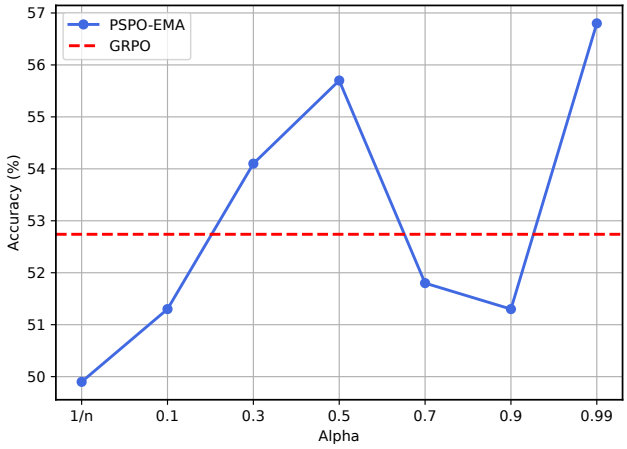

Figure 4: Average accuracy varies with alpha

The sample-average update $\alpha_n(q) = \frac{1}{n}$ ensures convergence in stationary settings (Sutton, Barto et al. 1998; Liang et al. 2021), but adapts poorly under nonstationary reward distributions. In RFT, where model behavior and preferences shift over time, emphasizing recent rewards is more effective.

Figure 4 shows average accuracy across five datasets; per-dataset results are in the appendix. The best performance occurs at $\alpha_n(q) = 0.99$, indicating that heavily weighting recent rewards with slight influence from the past offers the best balance between stability and adaptation. In contrast, The sample-average update rule $\alpha_n(q) = \frac{1}{n}$ yields the lowest performance among all tested configurations. This can be largely attributed to its assumption of a stationary reward distribution, which fails to hold in the RFT of LLMs. In such nonstationary environments, older reward signals become less relevant over time, and averaging them equally—as done in the $\frac{1}{n}$ scheme—hinders the model's ability to adapt to evolving reward dynamics. Consequently, it underperforms compared to constant step-size approaches that prioritize recent feedback. Interestingly, performance peaks again at $\alpha_n(q) = 0.5$, showing a good trade-off between stability and responsiveness.

Conclusion

We propose PSPO, a lightweight extension to GRPO that enhances training efficiency and stability for RFT. By incorporating ERS and PPS, PSPO improves gradient estimation and focuses training on high-impact prompts, all without significant computational or memory overhead. Extensive experiments on math reasoning benchmarks show that PSPO matches or surpasses GRPO and Dr.GRPO in performance while accelerating convergence. While effective, PSPO currently focuses only on advantage-based prioritization. Future work could explore alternative strategies to further boost sample efficiency. Notably, both components are compatible with other RL methods.

Acknowledgments

This work is supported in part by Shenzhen Science and Technology Program under Grant ZDSYS20220527171400002, the National Natural Science Foundation of China (NSFC) under Grants 62271324, 62231020, 62394335 and 62371309, and the Open Research Fund from Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ) (Grant No. GML-LF-24-32).

References

Ahmadian, A.; Cremer, C.; Gallé, M.; Fadaee, M.; Kreutzer, J.; Pietquin, O.; Üstün, A.; and Hooker, S. 2024. Back to Basics: Revisiting REINFORCE-Style Optimization for Learning from Human Feedback in LLMs. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 12248–12267. Bangkok, Thailand: Association for Computational Linguistics.

Albalak, A.; Elazar, Y.; Xie, S. M.; Longpre, S.; Lambert, N.; Wang, X.; Muennighoff, N.; Hou, B.; Pan, L.; Jeong, H.; Raffel, C.; Chang, S.; Hashimoto, T.; and Wang, W. Y. 2024. A Survey on Data Selection for Language Models. arXiv:2402.16827.

Aminabadi, R. Y.; Rajbhandari, S.; Zhang, M.; Awan, A. A.; Li, C.; Li, D.; Zheng, E.; Rasley, J.; Smith, S.; Ruwase, O.; and He, Y. 2022. DeepSpeed Inference: Enabling Efficient Inference of Transformer Models at Unprecedented Scale. arXiv:2207.00032.

Chen, X.; Zhu, W.; Qiu, P.; Dong, X.; Wang, H.; Wu, H.; Li, H.; Sotiras, A.; Wang, Y.; and Razi, A. 2025. Dragrpo: Exploring diversity-aware reward adjustment for r1-zero-like training of large language models. *arXiv preprint arXiv:2505.09655*.

Dang, Q.-A.; and Ngo, C. 2025. Reinforcement Learning for Reasoning in Small LLMs: What Works and What Doesn't. *arXiv preprint arXiv:2503.16219*.

Di-Castro, S.; Mannor, S.; and Di Castro, D. 2022. Analysis of stochastic processes through replay buffers. In *International Conference on Machine Learning*, 5039–5060. PMLR.

Dong, G.; Mao, H.; Ma, K.; Bao, L.; Chen, Y.; Wang, Z.; Chen, Z.; Du, J.; Wang, H.; Zhang, F.; Zhou, G.; Zhu, Y.; Wen, J.-R.; and Dou, Z. 2025. Agentic Reinforced Policy Optimization. arXiv:2507.19849.

Finch, T. 2009. Incremental calculation of weighted mean and variance. Technical report, University of Cambridge Computing Service. Accessed: 2025-07-24.

Gu, S. S.; Lillicrap, T.; Turner, R. E.; Ghahramani, Z.; Schölkopf, B.; and Levine, S. 2017. Interpolated policy gradient: Merging on-policy and off-policy gradient estimation for deep reinforcement learning. *Advances in neural information processing systems*, 30.

Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

He, C.; Luo, R.; Bai, Y.; Hu, S.; Thai, Z.; Shen, J.; Hu, J.; Han, X.; Huang, Y.; Zhang, Y.; Liu, J.; Qi, L.; Liu, Z.; and Sun, M. 2024. OlympiadBench: A Challenging Benchmark for Promoting AGI with Olympiad-Level Bilingual Multimodal Scientific Problems. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 3828–3850. Bangkok, Thailand: Association for Computational Linguistics.

Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021. Measuring Mathematical Problem Solving With the MATH Dataset. *NeurIPS*.

Hilton, J.; Cobbe, K.; and Schulman, J. 2022. Batch size-invariance for policy optimization. *Advances in Neural Information Processing Systems*, 35: 17086–17098.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2021. LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685.

Hu, J. 2025. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv preprint arXiv:2501.03262*.

Hugging Face. 2025. Open R1: A fully open reproduction of DeepSeek-R1.

Kaddour, J.; Key, O.; Nawrot, P.; Minervini, P.; and Kusner, M. J. 2023. No train no gain: Revisiting efficient training algorithms for transformer-based language models. *Advances in Neural Information Processing Systems*, 36: 25793–25818.

Kumar, K.; Ashraf, T.; Thawakar, O.; Anwer, R. M.; Cholakkal, H.; Shah, M.; Yang, M.-H.; Torr, P. H.; Khan, F. S.; and Khan, S. 2025. Llm post-training: A deep dive into reasoning large language models. *arXiv preprint arXiv:2502.21321*.

Lanchantin, J.; Chen, A.; Lan, J.; Li, X.; Saha, S.; Wang, T.; Xu, J.; Yu, P.; Yuan, W.; Weston, J. E.; et al. 2025. Bridging Offline and Online Reinforcement Learning for LLMs. *arXiv preprint arXiv:2506.21495*.

Lewkowycz, A.; Andreassen, A.; Dohan, D.; Dyer, E.; Michalewski, H.; Ramasesh, V.; Slone, A.; Anil, C.; Schlag, I.; Gutman-Solo, T.; et al. 2022. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35: 3843–3857.

Li, G.; Lin, M.; Galanti, T.; Tu, Z.; and Yang, T. 2025a. DisCO: Reinforcing Large Reasoning Models with Discriminative Constrained Optimization. *arXiv preprint arXiv:2505.12366*.

Li, S.; Zhou, Z.; Lam, W.; Yang, C.; and Lu, C. 2025b. RePO: Replay-Enhanced Policy Optimization. *arXiv preprint arXiv:2506.09340*.

Li, Z.; Xu, T.; Zhang, Y.; Lin, Z.; Yu, Y.; Sun, R.; and Luo, Z.-Q. 2023. Remax: A simple, effective, and efficient reinforcement learning method for aligning large language models. *arXiv preprint arXiv:2310.10505*.

Liang, X.; Ma, Y.; Feng, Y.; and Liu, Z. 2021. Ptr-ppo: Proximal policy optimization with prioritized trajectory replay. *arXiv preprint arXiv:2112.03798*.

Liao, M.; Xi, X.; Chen, R.; Leng, J.; Hu, Y.; Zeng, K.; Liu, S.; and Wan, H. 2025. Enhancing Efficiency and Exploration in Reinforcement Learning for LLMs. *arXiv preprint arXiv:2505.18573*.

Liu, Z.; Chen, C.; Li, W.; Qi, P.; Pang, T.; Du, C.; Lee, W. S.; and Lin, M. 2025. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*.

Lu, F.; Zhong, Z.; Liu, S.; Fu, C.-W.; and Jia, J. 2025. ARPO: End-to-End Policy Optimization for GUI Agents with Experience Replay. *arXiv preprint arXiv:2505.16282*.

MAA: American Mathematics Competitions. 2023. AMC. https://maa.org/. Accessed: 2025-07-04.

Mroueh, Y. 2025. Reinforcement Learning with Verifiable Rewards: GRPO's Effective Loss, Dynamics, and Success Amplification. *arXiv preprint arXiv:2503.06639*.

Naik, A.; Wan, Y.; Tomar, M.; and Sutton, R. S. 2024. Reward Centering. *Reinforcement Learning Journal*, 4: 1995–2016.

Noukhovitch, M.; Lavoie, S.; Strub, F.; and Courville, A. 2023. Language Model Alignment with Elastic Reset. In *Neural Information Processing Systems*.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744.

Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2023. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Rajbhandari, S.; Rasley, J.; Ruwase, O.; and He, Y. 2020. ZeRO: Memory Optimizations Toward Training Trillion Parameter Models. arXiv:1910.02054.

Ren, Y.; Zhang, H.; Yu, F. R.; Li, W.; Zhao, P.; and He, Y. 2025. Industrial Internet of Things With Large Language Models (LLMs): An Intelligence-Based Reinforcement Learning Approach. *IEEE Trans. Mobile Computing*, 24(5): 4136–4152.

Schaul, T.; Quan, J.; Antonoglou, I.; and Silver, D. 2015. Prioritized Experience Replay. Cite arxiv:1511.05952Comment: Published at ICLR 2016.

Schulman, J.; Moritz, P.; Levine, S.; Jordan, M. I.; and Abbeel, P. 2016. High-Dimensional Continuous Control Using Generalized Advantage Estimation. In Bengio, Y.; and LeCun, Y., eds., *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y.; Wu, Y.; et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Sun, H.; and van der Schaar, M. 2024. Inverse-rlignment: Inverse reinforcement learning from demonstrations for llm alignment. *arXiv preprint arXiv:2405.15624*.

Sun, Y.; Shen, J.; Wang, Y.; Chen, T.; Wang, Z.; Zhou, M.; and Zhang, H. 2025. Improving Data Efficiency for LLM Reinforcement Fine-tuning Through Difficulty-targeted Online Data Selection and Rollout Replay. *arXiv preprint arXiv:2506.05316*.

Sutton, R. S.; Barto, A. G.; et al. 1998. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge.

Swamy, G.; Choudhury, S.; Sun, W.; Wu, Z. S.; and Bagnell, J. A. 2025. All Roads Lead to Likelihood: The Value of Reinforcement Learning in Fine-Tuning. arXiv:2503.01067.

Vemprala, S. H.; Bonatti, R.; Bucker, A.; and Kapoor, A. 2024. Chatgpt for robotics: Design principles and model abilities. *Ieee Access*.

von Werra, L.; Belkada, Y.; Tunstall, L.; Beeching, E.; Thrush, T.; Lambert, N.; Huang, S.; Rasul, K.; and Gallouédec, Q. 2020. TRL: Transformer Reinforcement Learning. https://github.com/huggingface/trl.

Wang, J. T.; Wu, T.; Song, D.; Mittal, P.; and Jia, R. 2024. GREATS: Online Selection of High-Quality Data for LLM Training in Every Iteration. In Globerson, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J.; and Zhang, C., eds., *Advances in Neural Information Processing Systems*, volume 37, 131197–131223. Curran Associates, Inc.

Wang, Q.; Ke, J.; Ye, H.; Lin, Y.; Fu, Y.; Zhang, J.; Keutzer, K.; Xu, C.; and Chen, Y. 2025. Angles Don't Lie: Unlocking Training-Efficient RL Through the Model's Own Signals. *arXiv preprint arXiv:2506.02281*.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3): 229–256.

Xu, S.; Fu, W.; Gao, J.; Ye, W.; Liu, W.; Mei, Z.; Wang, G.; Yu, C.; and Wu, Y. 2024. Is dpo superior to ppo for llm alignment? a comprehensive study. *arXiv preprint arXiv:2404.10719*.

Yang, A.; Zhang, B.; Hui, B.; Gao, B.; Yu, B.; Li, C.; Liu, D.; Tu, J.; Zhou, J.; Lin, J.; Lu, K.; Xue, M.; Lin, R.; Liu, T.; Ren, X.; and Zhang, Z. 2024. Qwen2.5-Math Technical Report: Toward Mathematical Expert Model via Self-Improvement. *arXiv preprint arXiv:2409.12122*.

Yu, Q.; Zhang, Z.; Zhu, R.; Yuan, Y.; Zuo, X.; Yue, Y.; Dai, W.; Fan, T.; Liu, G.; Liu, L.; et al. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Yu, Z.; Das, S.; and Xiong, C. 2024. MATES: Model-Aware Data Selection for Efficient Pretraining with Data Influence Models. In *NeurIPS*.

Zheng, H.; Zhou, Y.; Bartoldson, B. R.; Kailkhura, B.; Lai, F.; Zhao, J.; and Chen, B. 2025. Act Only When It Pays: Efficient Reinforcement Learning for LLM Reasoning via Selective Rollouts. *arXiv preprint arXiv:2506.02177*.