From Data-Centric to Sample-Centric: Enhancing LLM Reasoning via Progressive Optimization

Xinjie Chen[1,2], **Minpeng Liao**[2*], **Guoxin Chen**[2], **Chengxi Li**[2]
Biao Fu[2], **Kai Fan**[2*], **Xinggao Liu**[1*]
[1]Zhejiang University [2]Alibaba Group Tongyi Lab
{xinjiechen, lxg}@zju.edu.cn
{minpeng.lmp, chenguoxin.cgx, xiji.lcx, fubiaobiao.fu, k.fan}@alibaba-inc.com

Abstract

Reinforcement learning with verifiable rewards (RLVR) has recently advanced the reasoning capabilities of large language models (LLMs). We investigate RLVR from a sample-centric perspective and introduce **LPPO** (**L**earning-**P**rogress and **P**refix-guided **O**ptimization), a framework of progressive optimization techniques. Our work addresses a critical question: how to best leverage a small set of trusted, high-quality demonstrations, rather than simply scaling up data volume. First, motivated by how hints aid human problem-solving, we propose **prefix-guided sampling**, an online data augmentation method that incorporates partial solution prefixes from expert demonstrations to guide the policy, particularly for challenging instances. Second, inspired by how humans focus on important questions aligned with their current capabilities, we introduce **learning-progress weighting**, a dynamic strategy that adjusts each training sample's influence based on model progression. We estimate sample-level learning progress via an exponential moving average of per-sample pass rates, promoting samples that foster learning and de-emphasizing stagnant ones. Experiments on mathematical-reasoning benchmarks demonstrate that our methods outperform strong baselines, yielding faster convergence and a higher performance ceiling, with these gains proving robust across diverse model architectures, scales, and reinforcement learning optimizers.

1 Introduction

Large Language Models (LLMs) have achieved significant advancements in complex reasoning, largely attributed to the paradigm of Reinforcement Learning with Verifiable Reward (RLVR) (Gao et al., 2024a; DeepSeek-AI et al., 2025; Kimi et al., 2025). RLVR employs verifiable rewards to effectively guide solution exploration, and its potential was notably highlighted when Deepseek-R1-zero (DeepSeek-AI et al., 2025) demonstrated a pathway to enhance LLM reasoning via RLVR without necessitating supervised fine-tuning (SFT). This has spurred a considerable volume of research focused on advancing RLVR methods. Current efforts focus primarily on data curation (Hu et al., 2025; Albalak et al., 2025; Li et al., 2025; Ye et al., 2025), reward design (Hu et al., 2025; Liu et al., 2025; Xie et al., 2025; Aggarwal and Welleck, 2025), or refinement of core RL algorithms (Liu et al., 2025; Yu et al., 2025; Yuan et al., 2025a; Hu, 2025) from foundational RL methods like PPO (Schulman et al., 2017), GRPO (Shao et al., 2024), and REINFORCE (Ahmadian et al., 2024). For example, Hu et al. (2025) and Yu et al. (2025) filter out samples with either excessively high or zero pass rates based on the roll-out results of each problem. Li et al. (2025) proposes a learning impact measurement to select more valuable samples for training. Yu et al. (2025) and Liu et al. (2025) make improvements on the biased optimization of sequence length in GRPO. Meanwhile, Yuan et al. (2025b) addresses issues in PPO via value pretraining and decoupled-GAE.

Despite recent advances, most existing methods either treat all training samples uniformly or rely on static heuristics, missing the opportunity to further exploit the potential of individual samples. While acquiring additional data can improve performance, data collection is often expensive or impractical. In such cases, it becomes critical to maximize the learning contribution from each available sample.

In contrast, human learners naturally focus more on challenging problems, and when faced with particularly difficult tasks, they seek hints or guidance from teachers or textbooks to acquire new techniques. As high-quality reasoning data become increasingly scarce, fully utilizing every available training instance is more important than ever. Thus, the central question extends from "How can we

*Corresponding authors.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2227–2242

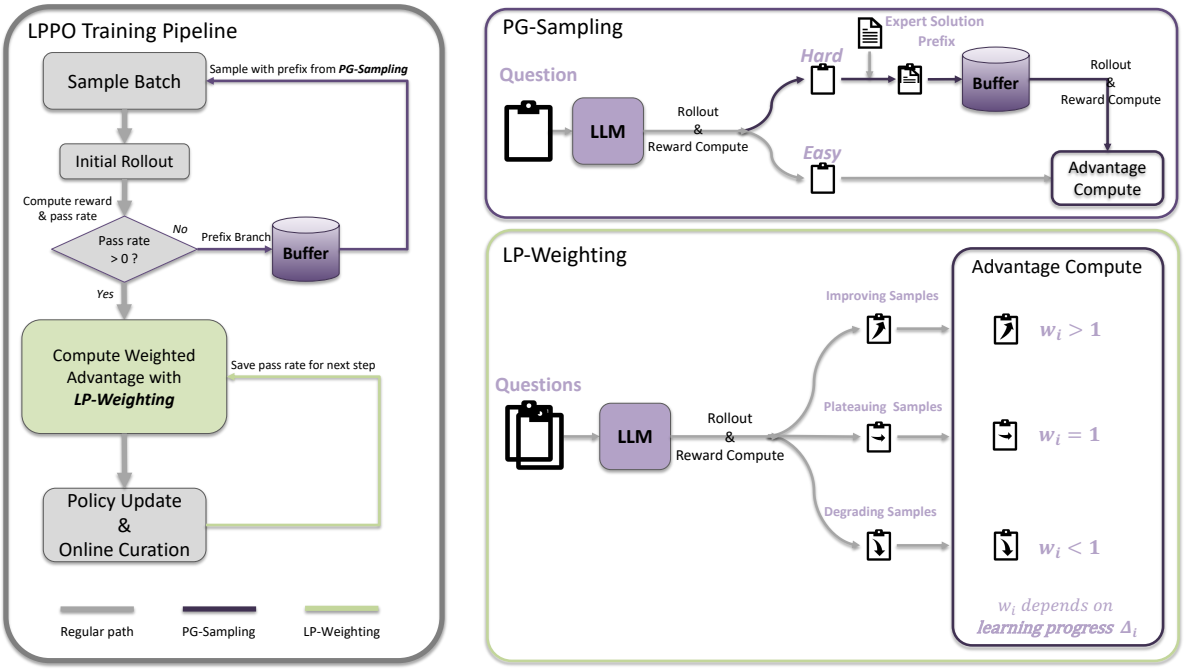

Figure 1: Overview of the proposed sample-centric RLVR framework. Left: the overall training pipeline. Each batch first undergoes an initial rollout to compute rewards and pass rates; samples with zero pass rate are revisited through prefix-guided augmentation, and sample-wise learning progress is used to compute weighted advantages for policy updates. Top right: PG-Sampling injects partial expert solution prefixes for hard questions to guide exploration. Bottom right: LP-Weighting up-weights improving samples, keeps plateauing samples near neutral, and down-weights degrading samples according to learning progress.

collect more data?" to "How can we best leverage a small set of trusted solutions, especially when the policy gets stuck?"

To address this, we extend our focus from a purely data-centric paradigm to also include a sample-centric perspective. Specifically, we propose Learning-Progress and Prefix-guided Optimization (LPPO), a sample-centric approach to reinforcement learning that aims to make the most of each solution-annotated sample, dynamically adjusting the optimization focus throughout training according to each sample's learning trajectory. Figure 1 provides an overview of this framework, illustrating how PG-Sampling and LP-Weighting are integrated into the RLVR training pipeline.

First, to better guide exploration on difficult problems, we propose **Prefix-Guided Sampling (PG-Sampling)**—a data augmentation technique that uses partial solution prefixes from expert models. Unlike supervised learning or behavior cloning, PG-Sampling mimics the process of learning from a hint rather than from complete solutions. Inspired by human learning, this approach allows the model to benefit from high-quality guidance while preserving the exploratory advantages of reinforcement learning, striking a balance between the limitations of supervised fine-tuning and the instability of pure RL.

Second, also motivated by the human learning process—where attention is often directed toward important questions in accordance with their current capabilities, we introduce **Learning-Progress Weighting (LP-Weighting)**, which dynamically adjusts each sample's influence based on the dynamics of RL training progress. Unlike uniform or static weighting, LP-Weighting tracks per-sample progress and prioritizes those where the model is improving, allocating resources more efficiently and accelerating convergence.

Our contributions can be summarized as follows:

- Inspired by the human learning process, we propose **LPPO** (**L**earning-**P**rogress and **P**refix-guided **O**ptimization), a novel sample-centric framework for RLVR that combines two complementary strategies: PG-Sampling and LP-Weighting.

- We demonstrate through comprehensive experiments on reasoning benchmarks that our approach, LPPO, outperforms strong RLVR

baselines.

- We empirically show that our approach achieves faster convergence and exhibits robust generalization, delivering consistent gains across diverse model sizes, architectures, and RL optimizers.

2 Related Work

2.1 RLVR in LLMs

Reinforcement Learning with Verifiable Reward (RLVR), where the reward is computed by a rule-based verification function, has been shown to be effective in improving the reasoning capabilities of LLMs. The most common practice of RLVR when applying reinforcement learning to LLMs on mathematical reasoning datasets is to use answer matching: the reward function outputs a binary signal based on whether the model's answer matches the gold reference answer (Gao et al., 2024b; DeepSeek-AI et al., 2025; Kimi et al., 2025; Zeng et al., 2025a; Wen et al., 2025; Song et al., 2025). This reward design obviates the need for complex outcome-based or process-based reward models, offering a straightforward yet potent approach. The efficacy of RLVR is further bolstered by algorithmic advancements in reinforcement learning. These include optimizations to value functions or policy updates within PPO (Schulman et al., 2017) (e.g., VinePPO (Kazemnejad et al., 2024), VCPPO (Yuan et al., 2025b), VAPO (Yue et al., 2025)), methods for stabilizing and accelerating GRPO (Shao et al., 2024) (e.g., DAPO (Yu et al., 2025), Dr. GRPO (Liu et al., 2025), SRPO (Zhang et al., 2025)), and the integration of diverse algorithmic components (e.g. REINFORCE++ (Hu, 2025)). Unlike these efforts, we focus on the temporal learning dynamics of individual samples within RLVR, which is an overlooked aspect.

2.2 Data Curation for LLM Post-Training

Data curation for LLM post-training has been extensively studied (Ivison et al., 2025), with a significant focus on strategies for supervised fine-tuning (SFT), also known as instruction tuning. These strategies encompass LLM-based quality assessments (Chen et al., 2024c), leveraging features derived from model computations (Ivison et al., 2023), gradient-based selection techniques (Xia et al., 2024), expert iteration based on Monte Carlo methods (Chen et al., 2024a; Zhang et al., 2024; Chen et al., 2024b), and knowledge distillation from expert model (Ye et al., 2025). Another line of research (Muldrew et al., 2024; Das et al., 2024) investigates data selection for human preference datasets within the Reinforcement Learning from Human Feedback (RLHF) paradigm (Ouyang et al., 2022). While data curation is well-established for SFT and RLHF, strategies specifically for RLVR are comparatively less explored. One notable attempt is LIMR (Li et al., 2025), which selected 1.4k examples from an 8.5k set for RLVR to match the performance of using the full set. Meanwhile, other studies (Fatemi et al., 2025; Wang et al., 2025) show that RLVR with very few examples can still improve performance, though with slower convergence. While insightful, most of these approaches focus on static data selection. More dynamic strategies, such as reverse curriculum reinforcement learning (Xi et al., 2024), offer a form of pre-set guided exploration, but leave online, sample-centric data augmentation for challenging samples largely unaddressed.

3 Methodology

3.1 From Data-Centric to Sample-Centric

While data-centric strategies are critical for LLM training and post-training efficiency (Kaplan et al., 2020; Hoffmann et al., 2022; Zha et al., 2025), their effectiveness is limited by the scarcity and cost of high-quality reasoning data. A sole reliance on acquiring more data thus encounters diminishing marginal returns, which motivates a pivot towards a "**sample-centric**" strategy. This approach aims to maximize learning efficacy from a given dataset by optimizing how each sample is used, a principle exemplified by methods like curriculum learning (Bengio et al., 2009; Narvekar et al., 2020), active learning (Xu et al., 2013), hard example mining (Smirnov et al., 2018), and sample weighting (Zhou et al., 2022). Given the data constraints in complex reasoning, we argue such a sample-level focus is crucial.

Within the RLVR framework, we therefore propose two sample-centric methods: (1) **Prefix-Guided Sampling**, a data augmentation strategy that utilizes partial expert solution prefixes to guide model exploration on challenging samples; and (2) **Learning-Progress Weighting**, a mechanism that adjusts the training influence of samples based on the model's learning progress on each.

3.2 Prefix-Guided Sampling

To better guide the exploration process, particularly for challenging problems, we introduce Prefix-Guided Sampling (PG-Sampling). This data augmentation strategy is inspired by partially observable reasoning trajectories. The technique involves guiding the policy by providing partial solutions as hints for difficult training samples. These prefixes are sampled from successful solutions generated by expert models.

Unlike RLVR using problem-answer datasets $D = \{q, a\}$, PG-Sampling requires datasets $D_{pg} = \{q, S_{\text{exp},q}, a\}$, where $S_{\text{exp},q}$ denotes the expert solution for a problem q. Thus, let Q_{sol} be the set of problems for which such expert solutions $S_{\text{exp},q}$ are available. From this set, a problem $q \in Q_{\text{sol}}$ is deemed "challenging" at a given training epoch t if its pass rate falls at or below a predefined threshold ϵ_c [1].

For a challenging problem $q \in Q_{\text{sol}}$, a prefix $S_{\text{pre},q}$ is generated from its expert solution $S_{\text{exp},q} = (y_1, y_2, \ldots, y_M)$, where M is the total number of tokens in the expert solution. The desired length of prefix, L_p, is determined by:

$$L_p = \lfloor \lambda \cdot M \rfloor \tag{1}$$

where λ is a truncation ratio randomly sampled from a uniform distribution $\mathcal{U}(\beta_{\min}, \beta_{\max})$ [2], $\lfloor \cdot \rfloor$ means L_p is further clipped to the last newline character to ensure the prefix ends at a complete line. The prefix $S_{\text{pre},q}$ then consists of the first L_p tokens of $S_{\text{exp},q}$:

$$S_{\text{pre},q} = (y_1, y_2, \ldots, y_{L_p}) \tag{2}$$

The policy π_θ then generates the remainder of the solution $S_{\text{rem},q}$ as follows:

$$S_{\text{rem},q} \sim \pi_\theta(\cdot | q \oplus S_{\text{pre},q}) \tag{3}$$

where $\oplus$ denotes token sequence concatenation. Specifically, the solution derived from $S_{\text{rem},q}$ is evaluated for the reward signal of RL training.

PG-Sampling guides the model using partial expert solutions, encouraging it to complete the remaining and thus balancing self-exploration with learning from examples. Unlike prior RLVR data curation methods that focused only on selection, PG-Sampling emphasizes augmenting and guiding challenging cases. While our approach of using partial trajectories shares a surface-level similarity with curriculum-based methods like Reverse Curriculum Reinforcement Learning (Xi et al., 2024), our online, sample-centric triggering mechanism is fundamentally different. A detailed comparison is provided in Section 2.2.

3.3 Learning-Progress Weighting

We introduce Learning-Progress Weighting (LP-Weighting), a method to dynamically re-weight each sample's *advantage estimate* in reinforcement learning. This re-weighting is implemented by applying a dynamic scaling factor, which is itself determined by the model's learning progress on that sample—specifically, its improvement (*e.g.*, pass rate) between adjacent epochs. The objective is to thereby amplify the influence of training samples where the model is actively improving and diminish that of samples where learning has stagnated or degraded.

For each sample (indexed by i), we track the Exponential Moving Average (EMA) (Hunter, 1986) of its pass rate at epoch t, denoted as $\overline{p_i}(t)$. The raw pass rate, $pass\_rate_i(t)$, derived from a finite number of rollouts, can exhibit high variance. The application of EMA helps to smooth these random fluctuations, providing a more stable assessment of the learning state:

$$\overline{p_i}(t) = \alpha \cdot pass\_rate_i(t) + (1-\alpha) \cdot \overline{p_i}(t-1) \tag{4}$$

where α is the smoothing factor. The initial $\overline{p_i}(0)$ is set to the pass rate observed at the first epoch.

Based on the smoothed pass rate, the learning progress for sample i at epoch t, denoted $\Delta_i(t)$, is defined as the first-order difference of its EMA pass rate:

$$\Delta_i(t) = \overline{p_i}(t) - \overline{p_i}(t-1) \tag{5}$$

This learning progress $\Delta_i(t)$ is then used to compute a dynamic weight $w_i(t)$ for sample i:

$$w_i(t) = \sigma(\kappa \cdot \Delta_i(t)) + b \tag{6}$$

where σ represents the sigmoid activation function, κ and b are factors that control the sensitivity and bias of the final weight to the learning progress.[3] The dynamic weight $w_i(t)$ is used during the RL

[1] $\epsilon_c = 0$ in our experiments, indicating that PG-Sampling is applied to problems the model fails to solve without guidance.

[2] We set $\beta_{\min} = 0.3, \beta_{\max} = 0.8$ in our experiments.

[3] We set κ=8.0 and b=0.5 in our experiments.

policy update phase to adjust the advantage estimate $\hat{A}_i$ for sample i. The weighted advantage, denoted as $\hat{A}'_i$, is calculated as:

$$\hat{A}'_i = w_i(t) \cdot \hat{A}_i \tag{7}$$

Alternatively, $w_i(t)$ can also be used as a sampling probability to determine if sample i is selected for the current training batch.

The LP-weighting method can be easily integrated with Group Relative Policy Optimization (GRPO) (Shao et al., 2024). The policy π_θ is optimized by maximizing the following objective:

$$\mathcal{J}_{\text{LP-GRPO}}(\theta) = \mathbb{E}\left[q_i \sim P(Q), \{o_{i,k}\}_{k=1}^{G} \sim \pi_{\theta_{\text{old}}}(O \mid q_i)\right] \left[\frac{1}{G}\sum_{k=1}^{G} \min\left(\rho_{i,k}(\theta)\hat{A}'_{i,k}, \quad \text{clip}(\rho_{i,k}(\theta), 1-\epsilon, 1+\epsilon)\hat{A}'_{i,k}\right)\right] \tag{8}$$

where $\rho_{i,k}(\theta) = \frac{\pi_\theta(o_{i,k}|q_i)}{\pi_{\theta_{old}}(o_{i,k}|q_i)}$ is the probability ratio for output $o_{i,k}$ of question q_i. For each question q_i (drawn from a data distribution of questions $P(Q)$), it performs rollouts with a group of size G, $\{o_{i,k}\}_{k=1}^{G}$ from the old policy $\pi_{\theta_{old}}(O \mid q_i)$. To encourage broader exploration, the KL penalty term is omitted from the objective.

In our context, the question q_i serves as the conditioning input to the policy π_θ; thus, the policy is conditioned on the actual content of question q_i, analogous to q in the original GRPO formulation. The term $\hat{A}'_{i,k}$ is the LP-weighted advantage for output $o_{i,k}$ of question q_i, calculated in the same way as Eq. (7).

Intuitively, if $\Delta_i(t) < 0$, it indicates that the policy rollouts for the i-th problem at epoch t perform worse than in the previous epoch. In such cases, we prefer not to emphasize these regressions during policy updates, avoiding reinforcement of "stagnant" or "degraded" solutions. Conversely, this mechanism, samples with positive learning progress (*i.e.*, $\Delta_i(t) > 0$) receive higher weights, correspondingly increasing their contribution to model parameter updates. To prevent the complete neglect of persistently challenging samples and to mitigate potential catastrophic forgetting, a minimum lower bound b is set for the weights $w_i(t)$, which is indicated in Eq. (6).

The LP-Weighting method aims to automatically direct the model's optimization focus towards samples where it is making substantial learning gains. This dynamic adjustment strategy is expected to enhance the overall efficiency and convergence performance of the training process, particularly when encountering training bottlenecks.

3.4 Detailed Training Algorithm

Integrating PG-Sampling and LP-Weighting into the RLVR training pipeline introduces several distinctions from the standard RLVR process. These are primarily centered around data preparation, batch construction, sample weighting, and online data curation:

Data Preparation The training data is categorized into two types: standard RLVR data denoted as D, and specialized data designated for PG-Sampling, D_{pg} (as detailed in Section 3.2). The D_{pg} dataset comprises challenging problems, each accompanied by an expert solution. It is also feasible to conduct training using only D_{pg}, in which case all problems processed during training are equipped with expert solutions.

Batch Construction with PG-Sampling At each training step, samples in the current batch undergo an initial rollout to calculate their pass rates. Samples with non-zero pass rates proceed with standard RL operations (e.g., advantage calculation, policy updates). Conversely, "difficult" samples from D_{pg} (those with zero pass rate) are augmented with a partial expert solution prefix and then stored for inclusion in the new batch for re-evaluation during the subsequent training step. This ensures these prefix-augmented difficult samples are revisited to maximize learning.

LP-Weighting Calculation For each sample i in a training batch, its pass rate $pass\_rate_i(t)$ is used to update its EMA pass rate $\overline{p_i}(t)$, subsequently determining the learning progress $\Delta_i(t)$ and the dynamic weight $w_i(t)$, as detailed in Section 3.3 (Eq. 4 to 6). During the policy update phase (e.g., using GRPO), these dynamic weights $w_i(t)$ are applied to the advantage values of their respective samples (Eq. 7).

Online Data Curation To maintain training efficiency and focus on more instructive examples, samples that achieve very high or very low pass rates will be excluded from the current batch and the next training epoch [4]. This adaptive data curation strategy allows the model to concentrate com-

[4]We exclude samples with a 100% or 0% pass rate during sampling, and exclude samples with a 100% pass rate for the next epoch in our experiments.

putational resources on more challenging or less frequently solved problems, thereby optimizing the learning process. We use this strategy in all our experiments, including the GRPO baseline.

The pseudocode of proposed algorithm is detailed in Appendix A.

4 Experiments

4.1 Experimental Setup

Datasets *Training Data:* Our training dataset is intentionally small and curated for quality. It comprises: 1. **The dataset with expert solutions**, denoted as D_{pg} in Section 3.2, consists of the same 817 examples as used in LIMO (Ye et al., 2025). These examples are high-quality mathematical problems selected from sources such as NuminaMath (Li et al., 2024) and historical AIME problems (1983–2023) (Veeraboina, 2023). They were specifically chosen based on their difficulty, generality, and diversity in required mathematical knowledge, thereby designed to elicit complex reasoning and accompanied by detailed step-by-step solutions provided by expert models (DeepSeek-AI et al., 2025; Huang et al., 2024). 2. **Level 3-5 problems from the MATH dataset** (Hendrycks et al., 2021), following the setup used in Liu et al. (2025). These slightly easier problems establish foundational skills and ensure broader competency, preventing over-specialization on the hardest tasks.

Evaluation Benchmarks: We evaluate on a diverse set of mathematical reasoning benchmarks: AIME24, AIME25, MATH-500 (Hendrycks et al., 2021; Lightman et al., 2024), AMC23 (Li et al., 2024), MinervaMath (Lewkowycz et al., 2022), and OlympiadBench (He et al., 2024).

Base Model To ensure consistency with prior studies, we use Qwen2.5-Math-7B (Yang et al., 2024) as our base model for RL fine-tuning, chosen for its strong mathematical reasoning abilities. To verify the generalizability of our methods, we also include experiments on different models, such as Qwen2.5-Base-14B and Llama-3.2-3B-Instruct.

Evaluation Metrics Following the settings in (DeepSeek-AI et al., 2025; Liu et al., 2025; Chu et al., 2025), the primary evaluation metric is *pass@1*, which is defined as the percentage of problems for which the model generates the correct final answer. To ensure fair comparisons, all reproduction results from third-party (Hochlehnert et al., 2025) or our own, which are tagged with † in Table 1, are averaged over three runs (*avg@3*). Detailed implementation settings are provided in Appendix B.

4.2 Main Results

Comparison Methods The proposed sample-centric methods are benchmarked against 7B models. Our direct RLVR baseline applies GRPO (Shao et al., 2024) to the Qwen2.5-Math-7B model (Yang et al., 2024). We further compare our results with several contemporary 7B models fine-tuned using RLVR without additional supervised fine-tuning beyond their initial pre-training, including Eurus-2-7B-PRIME (Cui et al., 2025), Oat-Zero-7B (Liu et al., 2025), OpenReasoner-Zero-7B (Hu et al., 2025), SimpleRL-Zero-MATH-7B (Zeng et al., 2025b), GPG-7B (Chu et al., 2025), and LIMR-7B (Li et al., 2025). For broader context, the performance of the base Qwen2.5-Math-7B and its instruction-tuned version, Qwen-2.5-Math-7B-Instruct, are also presented. For a fair and comprehensive comparison, results for external methods sourced from reproductions and original publications/papers are both presented.

Table 1 presents the main performance comparison on zero-shot *pass@1* across six mathematical reasoning benchmarks. Compared to the direct GRPO baseline, applying LP-Weighting alone consistently improves performance across all benchmarks, lifting the average score to 46.8% (+2.5%). This demonstrates the general effectiveness of dynamically adjusting sample influence based on learning progress. Integrating PG-Sampling on top of LP-Weighting yields further significant gains, achieving the highest average score of 48.8% among all listed models. This combined approach substantially outperforms the GRPO baseline (+4.5%) and also surpasses LP-Weighting alone (+2.0%), highlighting the strong additive benefit derived from using prefix guidance, particularly on challenging problems (e.g., boosting AIME24 score from 30.0% to 40.0%).

Our best model, LP-Weighting + PG-Sampling, also achieves the highest average performance among all strong contemporary 7B models. Specifically, it achieves the strongest results on AIME24, AIME25, and Minerva among our reproduced comparisons. While some specialized models show higher performance on individual benchmarks, our approach shows strong overall performance and generalization.

Model (7B)	AIME24	AIME25	AMC23	MATH-500	Minerva	OlympiadBench	Average	Samples (k)
Qwen-2.5-Math-7B-Instruct †	15.7	10.7	67.0	82.9	35.0	41.3	42.1	–
Qwen2.5-Math-7B (Base) †	20.7	8.7	56.2	64.3	17.3	29.0	32.7	–
Eurus-2–7B-PRIME	26.7	–	57.8	79.2	38.6	42.1	–	150
Eurus-2–7B-PRIME †	17.8	14.0	63.0	80.1	37.5	43.9	42.7	150
Oat-Zero-7B	43.3*	–	62.7	80.0	30.1	41.0	–	8.5
Oat-Zero-7B †	28.0	8.8	66.2	79.4	34.4	43.8	43.4	8.5
OpenReasoner-Zero-7B	17.9	15.6	–	81.4	–	–	–	129
OpenReasoner-Zero-7B †	19.7	15.7	59.5	**83.9***	31.6	**47.6***	43.0	129
SimpleRL-Zero-MATH-7B	24.0	–	70.0	80.2	37.5	39.0	–	8.5
SimpleRL-Zero-MATH-7B †	22.7	10.7	62.2	76.9	30.1	39.3	40.3	8.5
GPG-7B	33.3	–	65.0	80.0	34.2	42.4	–	17
GPG-7B †	26.7	10.0	**75.0***	79.8	39.3	44.7	45.9	17
LIMR-7B	32.5	–	63.8	78.0	–	–	–	1.4
LIMR-7B †	30.7	7.8	62.2	76.5	34.9	39.3	41.9	1.4
Qwen2.5-Math-7B + GRPO (Baseline)	30.0	10.0	62.5	80.7	39.4	43.1	44.3	9.2
+ LP-Weighting (Ours)	36.7	**16.7***	64.1	81.0	39.6	42.9	46.8	9.2
+ LP-Weighting + PG-Sampling (Our LPPO)	**40.0**	**16.7***	69.2	82.0	**41.5***	43.5	**48.8***	9.2

Table 1: Zero-shot *pass@1* performance of 7B models on six mathematical reasoning benchmarks. † indicates results obtained from reproductions. **Bolded** values denote the best reproducible results, and asterisked* values represent the best results among all reported results.

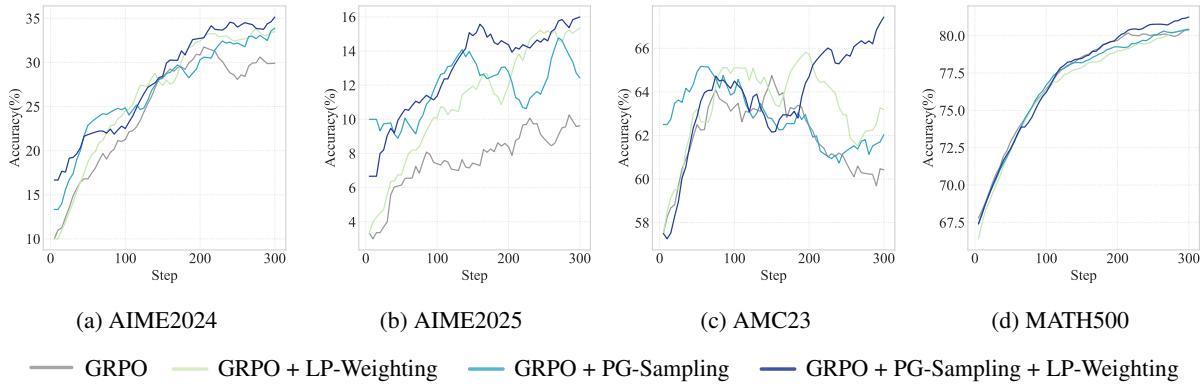

Figure 2: Ablation studies of our proposed LP-Weighting and PG-Sampling.

4.3 Ablation Studies

In Figure 2, we examine how different sample-centric methods influence the accuracy trajectories on four mathematical benchmarks throughout training. All curves are EMA-smoothed with $\alpha = 0.9$ and raw fluctuations are therefore slightly underestimated.

Early-stage acceleration (*Epoch 1*) During the first epoch (≤ 60 steps), the two strategies that include PG-Sampling dominate the baseline on every benchmark, while the LP-ONLY variant follows the baseline closely. Because LP-Weighting requires the pass rates from the previous epoch to compute dynamic advantages, its weights are still uniform at this stage; consequently, the optimization policies of PG and PG+LP are identical, and the visible oscillation stems solely from the randomly drawn reasoning prefixes and the training noise.

Mid-/late-stage gains (*Epoch 2+*) From step $\sim$60 onwards, the influence of LP-Weighting manifests as a steeper slope and lower variance: samples that achieve higher improvement on pass rates are up-weighted, which filters noisy gradients and, once a hard question starts to yield improvement, quickly amplifies its influence. The compound variant consequently outperforms all others, showing that the two techniques are complementary rather than redundant.

Building on above observations, we now examine how the two mechanisms steer learning dynamics: **Fast start.** PG-Sampling appends a solution prefix, giving an immediate accuracy jump and reducing exploration steps; **Reliable finish.** From epoch 2 onward, LP-Weighting shifts attention toward samples that raise pass rates, filtering gradient noise and lifting the ceiling. Used jointly, the two mechanisms drive the quick convergence and high accuracy on every benchmark.

4.4 Experiment Analysis

LP-Weighting Boosts Positive Sample Training Fig. 3(a) shows that models trained with LP-Weighting achieve higher average rewards over

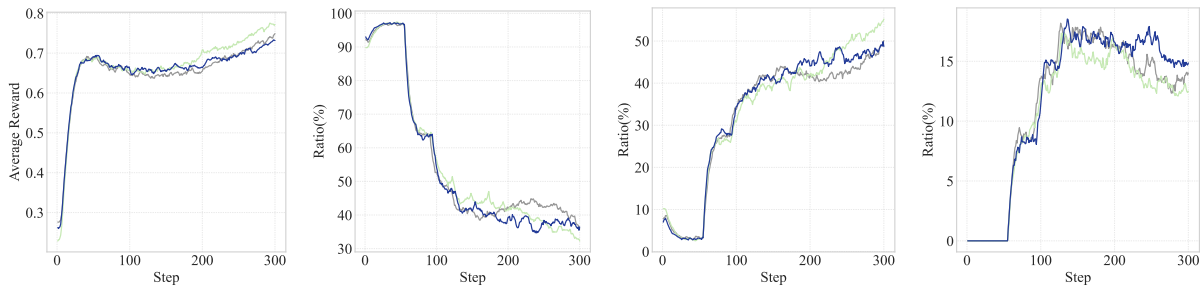

Figure 3: Training dynamics under GRPO: (a) average reward, and (b–d) proportions of improving, plateauing, and degrading samples for the baseline GRPO, GRPO + LP-Weighting, and GRPO with reversed LP-Weighting.

Model (7B)	AIME24	AIME25	AMC23	MATH500	Minerva	OlympiadBench	Average
			pass@1				
Qwen2.5-Math-7B + GRPO (Baseline)	30.0	10.0	62.5	80.7	39.4	43.1	44.3
+ LP-Weighting	36.7	**16.7**	64.1	81.0	39.6	42.9	46.8
+ LP-Weighting + PG-Sampling (β_{min}=0.1, β_{max}=0.5)	36.7	13.3	70.0	81.9	40.0	44.4	47.7
+ LP-Weighting + PG-Sampling (β_{min}=0.2, β_{max}=0.65)	36.7	13.3	65.8	83.4	41.9	43.7	47.5
+ LP-Weighting + PG-Sampling (β_{min}=0.3, β_{max}=0.8)	**40.0**	**16.7**	69.2	82.0	41.5	43.5	48.8
			pass@3				
Qwen2.5-Math-7B + GRPO (Baseline)	30.0	10.0	67.5	82.0	40.6	46.2	46.1
+ LP-Weighting	36.7	**16.7**	70.0	84.0	43.5	46.8	49.6
+ LP-Weighting + PG-Sampling (β_{min}=0.1, β_{max}=0.5)	36.7	13.3	**75.0**	83.2	42.3	**47.1**	49.6
+ LP-Weighting + PG-Sampling (β_{min}=0.2, β_{max}=0.65)	36.7	**16.7**	72.5	**84.4**	41.9	46.0	49.7
+ LP-Weighting + PG-Sampling (β_{min}=0.3, β_{max}=0.8)	**40.0**	**16.7**	72.5	83.8	**46.0**	46.6	**50.9**

Table 2: Comparison of model performance across six mathematical reasoning benchmarks. Results are reported using *pass@1* and *pass@3* metrics for different variants of PG-Sampling strategies.

time than both the GRPO baseline and the reversed-weighting variant, indicating more effective learning. Fig. 3(b) shows that under LP-Weighting the fraction of improving samples steadily decreases, suggesting the model converges as more samples are learned. Conversely, Fig. 3(c) shows that the proportion of plateauing samples (those with stable performance) rises with LP-Weighting, reflecting that many examples have reached a learning plateau. In Fig. 3(d), the proportion of degrading samples (those losing performance) falls under LP-Weighting, implying increased robustness and fewer regressions. Together, these results indicate that LP-Weighting helps the model retain high-pass-rate samples and accelerates training by prioritizing samples with active learning signals.

Diversity We examine how PG-Sampling affects the diversity of generated solutions using *pass@k* metrics. Table 2 reports both *pass@1* (*avg@3*) and *pass@3*. All methods achieve higher *pass@3* than *pass@1*, as expected, and PG-Sampling does not collapse this gap. In fact, LP-Weighting alone already widens the *pass@3* margin over the baseline, and adding prefix-guided sampling maintains or even slightly enlarges it. For example, the $(0.3, 0.8)$ PG-Sampling setting boosts *pass@3* in line with *pass@1*, indicating that multiple distinct correct answers remain available. In short, PG-Sampling preserves rich solution diversity: its accuracy gains come with sustained improvements in *pass@3*, showing that the model continues to generate a variety of valid solutions for each problem.

Impact of β We investigate how the truncation ratio range $(\beta_{\min}, \beta_{\max})$ in PG-Sampling affects accuracy. As shown in Table 2, all PG-Sampling variants with LP-Weighting outperform both the RLVR baseline and LP-Weighting alone. Expanding the prefix range generally improves results, with the widest range $(0.3, 0.8)$ achieving the best performance. Narrower ranges still help but less so. This indicates that longer prefixes better guide the model, though gains taper off at extremes. In practice, a moderate-to-large β range provides the best balance between effectiveness and exploration.

Impact of κ We sweep the slope multiplier $\kappa \in \{4, 8, 12\}$, analytically fixing the bias at $b = 0.5$ so that the expected weight $\mathbb{E}[w_i]$ in

Method / κ	AIME24	AIME25	AMC23	MATH-500	Minerva	OlympiadBench	Avg
Qwen2.5-Math-7B + GRPO (Baseline)	30.0	10.0	62.5	80.7	39.4	43.1	44.3
+ PG-Sampling + LP-Weighting (κ=4)	**43.3**	**16.7**	65.0	82.2	**42.2**	**44.6**	**49.0**
+ PG-Sampling + LP-Weighting (κ=8)	40.0	**16.7**	**69.2**	82.0	41.5	43.5	48.8
+ PG-Sampling + LP-Weighting (κ=12)	40.0	13.3	68.3	**82.8**	41.5	44.0	48.3

Table 3: Effect of the LP-Weighting scaling factor κ on pass@1 metric.

Setting	Baseline	+LPPO	Δ (pp)
Qwen-2.5-14B + GRPO	42.7	45.0	+2.3
Llama-3.2-3B-Instruct + GRPO	25.3	27.9	+2.6
Qwen-2.5-Math-7B + REINFORCE++	44.6	48.7	+4.1

Table 4: Robustness of LPPO across scale, architecture, and learner. Without extra tuning, LPPO consistently boosts *pass@1* over larger parameter budgets, alternative backbones, and REINFORCE-style learners.

Eq.6 is centred around 1—preventing either easy or hard samples from being systematically over- or under-emphasised. As Table 3 shows, every LP-Weighting variant surpasses the RLVR baseline, and the macro average changes by at most 0.7 pp, demonstrating that LP-Weighting is largely insensitive to this hyper-parameter.

LPPO under Diverse Scenarios. To verify that our LPPO generalises beyond the original Qwen-2.5-Math-7B setup, we replicate RLVR training in three orthogonal settings without retuning any hyper-parameters: (i) a larger backbone with Qwen-2.5-14B, (ii) a different backbone family using Llama-3.2-3B-Instruct, and (iii) an alternative policy-gradient learner (REINFORCE++). Table 4 summarises the results, while per-benchmark breakdowns are deferred to Appendix E. Across all variants, LPPO consistently yields +2–4 pp absolute improvements in *pass@1*, demonstrating robustness to model scale, backbone architecture, and RL optimiser choice.

5 Conclusion

We present **LPPO**, a sample-centric framework for RLVR. LPPO unites **LP-weighting** to focus on examples that drive improvement with **PG-Sampling** to provide hints for problems the policy cannot solve. By concentrating computation and injecting guidance on demand, LPPO shortens training time and consistently outperforms all baselines. Furthermore, it delivers consistent 2–4 pp pass@1 accuracy gains across diverse backbones, architectures, and learners. These results confirm that fine-grained, sample-level control offers a practical route to stronger mathematical reasoning LLMs.

Limitations

While our LPPO demonstrates clear advantages on mathematical reasoning benchmarks, several limitations remain. First, our experiments are conducted primarily on mathematical reasoning tasks with relatively small, high-quality expert-annotated datasets; the generalizability of these methods to broader reasoning domains or more diverse tasks requires further investigation. Second, PG-Sampling relies on the availability of expert solutions for challenging problems, which may not always be feasible for other tasks. Lastly, the overall improvements are bounded by the underlying model capacity and the difficulty of the evaluation benchmarks, and may not directly translate to real-world applications. We leave these directions for future work.

References

Pranjal Aggarwal and Sean Welleck. 2025. L1: controlling how long A reasoning model thinks with reinforcement learning. *CoRR*, abs/2503.04697.

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. 2024. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 12248–12267. Association for Computational Linguistics.

Alon Albalak, Duy Phung, Nathan Lile, Rafael Rafailov, Kanishk Gandhi, Louis Castricato, Anikait Singh, Chase Blagden, Violet Xiang, Dakota Mahan, and Nick Haber. 2025. Big-math: A large-scale, high-quality math dataset for reinforcement learning in language models. *Preprint*, arXiv:2502.17387.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, volume 382 of *ACM International Conference Proceeding Series*, pages 41–48. ACM.

Guoxin Chen, Minpeng Liao, Chengxi Li, and Kai Fan. 2024a. Alphamath almost zero: Process supervision

without process. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Guoxin Chen, Minpeng Liao, Chengxi Li, and Kai Fan. 2024b. Step-level value preference optimization for mathematical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, Findings of ACL, pages 7889–7903. Association for Computational Linguistics.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. 2024c. Alpagasus: Training a better alpaca with fewer data. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Xiangxiang Chu, Hailang Huang, Xiao Zhang, Fei Wei, and Yong Wang. 2025. Gpg: A simple and strong reinforcement learning baseline for model reasoning. *Preprint*, arXiv:2504.02546.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, Hao Peng, Yu Cheng, and 4 others. 2025. Process reinforcement through implicit rewards. *CoRR*, abs/2502.01456.

Nirjhar Das, Souradip Chakraborty, Aldo Pacchiano, and Sayak Ray Chowdhury. 2024. Active preference optimization for sample efficient rlhf. *Preprint*, arXiv:2402.10500.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, and et al Xiao Bi. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948.

Mehdi Fatemi, Banafsheh Rafiee, Mingjie Tang, and Kartik Talamadupula. 2025. Concise reasoning via reinforcement learning. *Preprint*, arXiv:2504.05185.

Jiaxuan Gao, Shusheng Xu, Wenjie Ye, Weilin Liu, Chuyi He, Wei Fu, Zhiyu Mei, Guangju Wang, and Yi Wu. 2024a. On designing effective RL reward at training time for LLM reasoning. *CoRR*, abs/2410.15115.

Jiaxuan Gao, Shusheng Xu, Wenjie Ye, Weilin Liu, Chuyi He, Wei Fu, Zhiyu Mei, Guangju Wang, and Yi Wu. 2024b. On designing effective RL reward at training time for LLM reasoning. *CoRR*, abs/2410.15115.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 2024. Olympiadbench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 3828–3850. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Andreas Hochlehnert, Hardik Bhatnagar, Vishaal Udandarao, Samuel Albanie, Ameya Prabhu, and Matthias Bethge. 2025. A sober look at progress in language model reasoning: Pitfalls and paths to reproducibility. *Preprint*, arXiv:2504.07086.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, and 3 others. 2022. Training compute-optimal large language models. *CoRR*, abs/2203.15556.

Jian Hu. 2025. REINFORCE++: A simple and efficient approach for aligning large language models. *CoRR*, abs/2501.03262.

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. 2025. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *Preprint*, arXiv:2503.24290.

Zhen Huang, Haoyang Zou, Xuefeng Li, Yixiu Liu, Yuxiang Zheng, Ethan Chern, Shijie Xia, Yiwei Qin, Weizhe Yuan, and Pengfei Liu. 2024. O1 replication journey - part 2: Surpassing o1-preview through simple distillation, big progress or bitter lesson? *CoRR*, abs/2411.16489.

J Stuart Hunter. 1986. The exponentially weighted moving average. *Journal of quality technology*, 18(4):203–210.

Hamish Ivison, Noah A. Smith, Hannaneh Hajishirzi, and Pradeep Dasigi. 2023. Data-efficient finetuning using cross-task nearest neighbors. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 9036–9061. Association for Computational Linguistics.

Hamish Ivison, Muru Zhang, Faeze Brahman, Pang Wei Koh, and Pradeep Dasigi. 2025. Large-scale data selection for instruction tuning. *CoRR*, abs/2503.01807.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *CoRR*, abs/2001.08361.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron C. Courville, and Nicolas Le Roux. 2024. Vineppo: Unlocking RL potential for LLM reasoning through refined credit assignment. *CoRR*, abs/2410.01679.

Kimi, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, Chuning Tang, Congcong Wang, Dehao Zhang, Enming Yuan, Enzhe Lu, Fengxiang Tang, Flood Sung, Guangda Wei, Guokun Lai, and 75 others. 2025. Kimi k1.5: Scaling reinforcement learning with llms. *CoRR*, abs/2501.12599.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles, SOSP 2023, Koblenz, Germany, October 23-26, 2023*, pages 611–626. ACM.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay V. Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, and 1 others. 2024. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13:9.

Xuefeng Li, Haoyang Zou, and Pengfei Liu. 2025. LIMR: less is more for RL scaling. *CoRR*, abs/2502.11886.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. Understanding r1-zero-like training: A critical perspective. *CoRR*, abs/2503.20783.

William Muldrew, Peter Hayes, Mingtian Zhang, and David Barber. 2024. Active preference learning for large language models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Sanmit Narvekar, Bei Peng, Matteo Leonetti, Jivko Sinapov, Matthew E. Taylor, and Peter Stone. 2020. Curriculum learning for reinforcement learning domains: A framework and survey. *J. Mach. Learn. Res.*, 21:181:1–181:50.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. Hybridflow: A flexible and efficient RLHF framework. In *Proceedings of the Twentieth European Conference on Computer Systems, EuroSys 2025, Rotterdam, The Netherlands, 30 March 2025 - 3 April 2025*, pages 1279–1297. ACM.

Evgeny Smirnov, Aleksandr Melnikov, Andrei Oleinik, Elizaveta Ivanova, Ilya Kalinovskiy, and Eugene Luckyanets. 2018. Hard example mining with auxiliary embeddings. In *2018 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 37–46. Computer Vision Foundation / IEEE Computer Society.

Mingyang Song, Mao Zheng, Zheng Li, Wenjie Yang, Xuan Luo, Yue Pan, and Feng Zhang. 2025. Fastcurl: Curriculum reinforcement learning with progressive context extension for efficient training r1-like reasoning models. *CoRR*, abs/2503.17287.

Hemish Veeraboina. 2023. Aime problem set 1983–2024. https://www.kaggle.com/datasets/hemishveeraboina/aime-problem-set-1983-2024.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Lucas Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, Weizhu Chen, Shuohang Wang, Simon Shaolei Du, and Yelong Shen. 2025. Reinforcement learning for reasoning in large language models with one training example. *Preprint*, arXiv:2504.20571.

Liang Wen, Yunke Cai, Fenrui Xiao, Xin He, Qi An, Zhenyu Duan, Yimin Du, Junchen Liu, Lifu Tang, Xiaowei Lv, Haosheng Zou, Yongchao Deng, Shousheng Jia, and Xiangzheng Zhang. 2025. Light-r1: Curriculum sft, DPO and RL for long COT from scratch and beyond. *CoRR*, abs/2503.10460.

Zhiheng Xi, Wenxiang Chen, Boyang Hong, Senjie Jin, Rui Zheng, Wei He, Yiwen Ding, Shichun Liu, Xin Guo, Junzhe Wang, Honglin Guo, Wei Shen, Xiaoran Fan, Yuhao Zhou, Shihan Dou, Xiao Wang, Xinbo Zhang, Peng Sun, Tao Gui, and 2 others. 2024. Training large language models for reasoning through reverse curriculum reinforcement learning. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. LESS: selecting influential data for targeted instruction tuning. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Tian Xie, Zitian Gao, Qingnan Ren, Haoming Luo, Yuqian Hong, Bryan Dai, Joey Zhou, Kai Qiu, Zhirong Wu, and Chong Luo. 2025. Logic-rl: Unleashing LLM reasoning with rule-based reinforcement learning. *CoRR*, abs/2502.14768.

Yan Xu, Fuming Sun, and Xue Zhang. 2013. Literature survey of active learning in multimedia annotation and retrieval. In *International Conference on Internet Multimedia Computing and Service, ICIMCS '13, Huangshan, China - August 17 - 19, 2013*, pages 237–242. ACM.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *CoRR*, abs/2409.12122.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. 2025. LIMO: less is more for reasoning. *CoRR*, abs/2502.03387.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, and 16 others. 2025. DAPO: an open-source LLM reinforcement learning system at scale. *CoRR*, abs/2503.14476.

Yufeng Yuan, Yu Yue, Ruofei Zhu, Tiantian Fan, and Lin Yan. 2025a. What's behind ppo's collapse in long-cot? value optimization holds the secret. *CoRR*, abs/2503.01491.

Yufeng Yuan, Yu Yue, Ruofei Zhu, Tiantian Fan, and Lin Yan. 2025b. What's behind ppo's collapse in long-cot? value optimization holds the secret. *CoRR*, abs/2503.01491.

Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, Xiangpeng Wei, Xiangyu Yu, Gaohong Liu, Juncai Liu, Lingjun Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Chi Zhang, and 8 others. 2025. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks. *Preprint*, arXiv:2504.05118.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. 2025a. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *CoRR*, abs/2503.18892.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. 2025b. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *CoRR*, abs/2503.18892.

Daochen Zha, Zaid Pervaiz Bhat, Kwei-Herng Lai, Fan Yang, Zhimeng Jiang, Shaochen Zhong, and Xia Ben Hu. 2025. Data-centric artificial intelligence: A survey. *ACM Comput. Surv.*, 57(5):129:1–129:42.

Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. 2024. Rest-mcts*: LLM self-training via process reward guided tree search. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Xiaojiang Zhang, Jinghui Wang, Zifei Cheng, Wenhao Zhuang, Zheng Lin, Minglei Zhang, Shaojie Wang, Yinghan Cui, Chao Wang, Junyi Peng, and 1 others. 2025. Srpo: A cross-domain implementation of large-scale reinforcement learning on llm. *arXiv preprint arXiv:2504.14286*.

Xiaoling Zhou, Ou Wu, Weiyao Zhu, and Ziyang Liang. 2022. Understanding difficulty-based sample weighting with a universal difficulty measure. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2022, Grenoble, France, September 19-23, 2022, Proceedings, Part III*, volume 13715 of *Lecture Notes in Computer Science*, pages 68–84. Springer.

Algorithm 1 Integrated Training Process with Proposed LPPO

Require: Standard RLVR dataset $\mathcal{D}$, PG-sampling subset $\mathcal{D}_{pg}$, initial policy parameters θ, EMA decay α, LP-weighting sensitivity κ, LP-weighting bias b
Ensure: Updated policy parameters θ

// 1. Initialize EMA of pass rates
Initialize EMA of pass rates $\overline{p_i}(0) \leftarrow 0$ for all samples i
for epoch $t = 1$ to T **do**
  // 2. Compute pass rates
  for each sample i in $\mathcal{D} \cup \mathcal{D}_{pg}$ **do**
    Roll out current policy π_θ on sample i to compute $\text{pass\_rate}_i(t)$
    Compute reward r_i for trajectories from sample i
    if $0 < \text{pass\_rate}_i(t) < 1$ **then**
      Add these trajectories (with reward r_i) to the $batch$ for policy update
    end if
  end for
  // 3. PG-Sampling for difficult samples
  for each sample i in $\mathcal{D}_{pg}$ with $\text{pass\_rate}_i(t) = 0$ **do**
    Generate a prefix $S_{\text{pre},i}$ for sample i according to Eq. 1 and Eq. 2
    Generate the remaining sequence (suffix) $S_{\text{rem},i}$ for this prefix using Eq. 3
    Compute reward r_i for trajectories from sample i
    Add these prefix-guided trajectories ($S_{\text{rem},i}$, with reward r_i) to the $batch$ for policy update
  end for
  // 4. Update pass-rate statistics and LP weights
  for each sample i in $batch$ **do**
    Update EMA of pass rate: $\overline{p_i}(t) \leftarrow (1-\alpha)\overline{p_i}(t-1) + \alpha\, \text{pass\_rate}_i(t)$ ▷ (Corresponds to Eq. 4)
    Compute learning progress: $\Delta_i(t) \leftarrow \overline{p_i}(t) - \overline{p_i}(t-1)$ ▷ (Corresponds to Eq. 5)
    Compute LP-weight: $w_i(t) \leftarrow \sigma(\kappa \cdot \Delta_i(t)) + b$ ▷ (Corresponds to Eq. 6, where σ is the sigmoid function)
  end for
  // 5. Compute weighted advantages
  for each sample i and rollout index k **do**
    Weight the advantage: $\hat{A}'_{i,k} \leftarrow w_i(t)\, \hat{A}_{i,k}$ ▷ (Corresponds to Eq. 7)
  end for
  // 6. Policy update
  Update policy π_θ via GRPO ▷ (Corresponds to Eq. 8)
  // 7. Online data curation
  Remove (or skip) consistently solved samples (with $\text{pass\_rate}_i(t) = 1.0$) in $\mathcal{D} \cup \mathcal{D}_{pg}$ from training
end for

Prompt

```
system
Please reason step by step, and put
your final answer within \\boxed{}.
user
{question}
assistant
{answer}
```

Figure 4: Prompt template.

Prompt For PG-Sampling

```
system
Please reason step by step, and put
your final answer within \\boxed{}.
user
{question}
assistant
{solution_prefix} + {answer}
```

Figure 5: Prompt template for PG-Sampling.

A Integrated Pipeline with Our LPPO

This section details the integrated progressive optimization RLVR pipeline that combines PG-Sampling and LP-Weighting to improve sample efficiency and training dynamics. The full algorithm is summarized in Algorithm A.

B Additional Experiment Settings

Implementation Details We use the verl (Sheng et al., 2025) pipeline for RL fine-tuning and evaluation. By default, the coefficients for entropy loss is set to -0.001. For training rollouts generated via vLLM (Kwon et al., 2023), we set the sampling temperature to 1.0. The optimizer uses a learning rate of 1×10^{-6} and a weight decay coefficient of 0.01. We use a training batch size of

128, a mini-batch size of 64, and generate 32 rollouts per sample. To accommodate potentially long sequences required by PG-Sampling, we set the maximum prompt lengths to 5000 tokens each, and the maximum response length is also 5000. Considering that the Qwen2.5-Math-7B model has only a 4096 context length by default, we expand this to 10,000. We store the model checkpoint every 5 steps for evaluation. Each experiment is trained for 300 steps on 8 A100 GPUs. The details of prompt and reward design are shown in Appendix B.

Prompt We use the same template as Qwen-MATH [5], shown in Fig. 4, for our training and evaluation. For PG-sampling, which combines the solution and the problem for difficult questions, we inject the prefix `solution_prefix` into the prompt for training, as shown in Fig. 5.

Reward Design We focus exclusively on the model's reasoning ability, not on exposing its entire chain of thought; hence we do not adopt the format reward for Deepseek-R1 (DeepSeek-AI et al., 2025). To minimise the risk of reward-hacking that can arise from elaborate scoring schemes, we employ a deliberately minimal, rule-based metric.

The sole component is mathematical correctness, denoted R_{math}; no extra credit is awarded for formatting or auxiliary details. After the model produces its final answer, an automatic verifier checks equivalence with the ground truth:

$$R_{\text{math}} = \begin{cases} 1, & \text{if the answer is completely correct,} \\ 0, & \text{otherwise.} \end{cases}$$

Accordingly, the total reward is simply

$$R = R_{\text{math}}.$$

C Efficiency Gains via Online Data Curation

Online data curation dynamically excludes both solved and overly difficult samples, allowing computation to focus on informative ones. Each useful sample is revisited more frequently within the same wall-clock time, amplifying the effect of LP-Weighting. As illustrated in Fig. 6, this process narrows the active set over time compared to the baseline, leading to faster convergence and improved synergy with LP-Weighting.

[5]https://huggingface.co/Qwen/Qwen2.5-Math-7B-Instruct

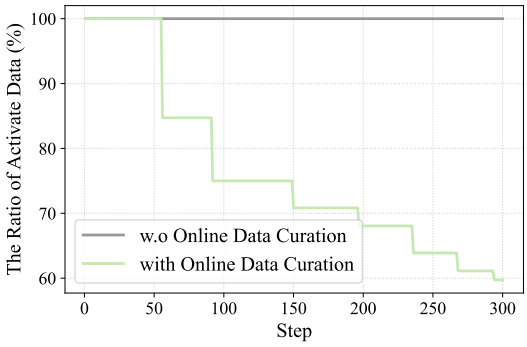

Figure 6: The Ratio of Active Data during Training.

D Further Exploration on PG-Sampling

KL Divergence Although not optimized, the KL divergence between the policy and reference models is monitored to gauge exploration. As shown in Fig. 7, the PG-Sampling variant diverges from the reference distribution more rapidly after around 150 updates, indicating that injecting a prefix after the question encourages greater exploration. After step 250, both curves reflect similar distances from the reference, but not necessarily similar policies. Combining the results in Fig. 2 and Fig. 7, we observe that although the PG-Sampling strategy does not further increase exploration divergence, its performance improves significantly. This indicates that PG-Sampling utilizes the exploration budget more efficiently and guides the policy toward higher-value regions of the solution space.

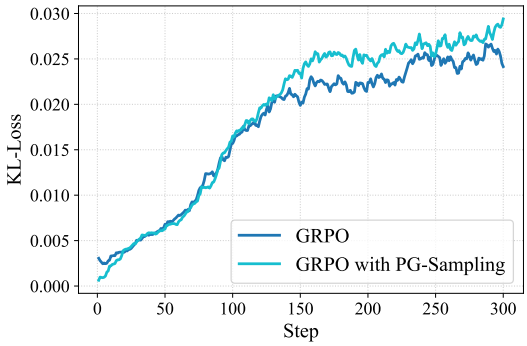

Figure 7: The KL Divergence with/without PG-Sampling (EMA smoothed with α=0.9).

Ratio of PG-Sampling Augmented Samples Figure 8 shows the ratio of PG-Sampling augmented samples during training. Due to the presence of online data curation, the proportion of samples affected by the PG-Sampling strategy gradually increases as the training proceeds, rising

steadily from an initial value of around 6% and eventually reaching approximately 11% at step 300. This gradual growth occurs because online data curation continuously filters out samples that become too easy, leaving behind those challenging instances that trigger the PG-Sampling augmentation. As a result, the proportion of prefix-guided samples within each training batch gradually increases over time.

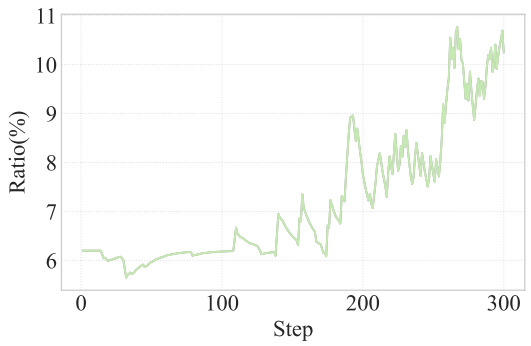

Figure 8: Ratio of PG-Sampling augmented samples over training steps.

Does PG-Sampling Induce Expert-Like Reasoning? The PG-Sampling strategy incorporates expert-generated partial solutions as inputs, potentially prompting the model to imitate the expert's reasoning pattern even without explicit SFT loss. To examine whether this sampling technique leads to outputs that closely align with expert solutions, we use the all-MiniLM-L6-v2 model (Wang et al., 2020) to measure the semantic similarity between the model-generated and oracle expert solutions.

Figure 9 illustrates the cosine similarity between the embeddings of model-generated solutions and oracle expert solutions, comparing training with and without PG-Sampling. Notably, the similarity curves for both methods exhibit similar trends and convergence behaviors. This observation indicates that incorporating expert solution prefixes does not inherently make the model's generated outputs significantly closer to expert reasoning styles. Consequently, the primary benefit of PG-Sampling is enhancing exploration efficiency rather than promoting imitation of expert solutions.

Lexical Marker Analysis. We further examine whether PG-Sampling simply induces imitation of the surface lexical style of expert solutions. Specifically, we compare the relative frequencies of selected discourse and reflective markers in expert solutions from the training set and in model-generated outputs from the baseline and LPPO models. As shown in Table 5, the baseline and LPPO models exhibit highly similar marker distributions, and both differ markedly from the expert solutions. In particular, markers such as *wait*, *maybe*, *perhaps*, and *let me*, which appear in expert solutions, are absent from both model outputs. These results complement the semantic-similarity analysis above and suggest that the gains of PG-Sampling are not primarily driven by imitation of the surface lexical style of expert solutions, but rather by more effective exploration during RL training.

E LPPO under Diverse Scenarios

This appendix complements Section 4.4 in the main text; the complete numerical results are listed in Table 6 for easy cross-reference.

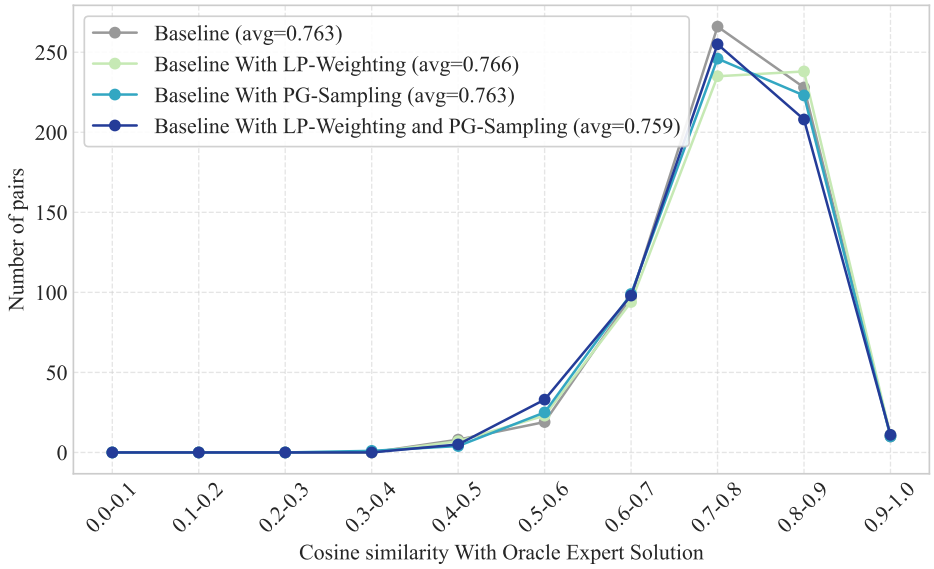

Figure 9: The Similarity between Oracle Expert Solution with/without PG-Sampling.

Marker	Expert (%)	Baseline (%)	LPPO (%)
so	27.75	32.23	31.36
but	21.59	3.31	2.54
wait	12.33	0.00	0.00
therefore	10.57	10.74	11.02
if	7.93	22.31	24.58
let's	5.73	28.93	27.97
let me	5.29	0.00	0.00
maybe	3.96	0.00	0.00
perhaps	2.64	0.00	0.00
because	2.20	2.48	2.54
Total	100.00	100.00	100.00

Table 5: Relative frequencies of selected discourse and reflective markers in expert solutions from the training set and in model-generated outputs from the baseline and LPPO models. Percentages are normalized within the selected marker set for each source. The close similarity between the baseline and LPPO outputs, together with their clear differences from the expert solutions, provides additional evidence that LPPO does not merely imitate the surface lexical style of expert demonstrations.

Setting	AIME24	AIME25	AMC23	MATH-500	Minerva	Olympiad	Avg.
Larger backbone: Qwen-2.5-14B							
Baseline (GRPO)	**13.3**	13.3	57.5	79.9	**47.3**	44.7	42.7
+LPPO	**13.3**	**20.0**	**62.5**	**82.2**	46.0	**46.1**	**45.0**
Different backbone: Llama-3.2-3B-Instruct							
Baseline (GRPO)	16.7	3.3	25.8	**58.0**	24.9	**23.1**	25.3
+LPPO	**20.0**	**6.7**	**35.0**	57.8	**25.0**	23.0	**27.9**
Different learner: Qwen-2.5-Math-7B + REINFORCE++							
Baseline	26.7	10.0	60.8	**81.2**	45.3	43.7	44.6
+LPPO	**43.3**	**13.3**	**65.0**	81.0	**45.5**	**44.0**	**48.7**

Table 6: Pass@1 accuracy (%) of LPPO across diverse scenarios. LPPO consistently brings +2–4 pp absolute improvements over each corresponding baseline without any hyper-parameter retuning.