BUFFER MATTERS: UNLEASHING THE POWER OF OFF-POLICY REINFORCEMENT LEARNING IN LARGE LANGUAGE MODEL REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Traditional on-policy Reinforcement Learning with Verifiable Rewards (RLVR) frameworks suffer from experience waste and reward homogeneity, which directly hinders learning efficiency on difficult samples during large language models post-training. In this paper, we introduce Batch Adaptation Policy Optimization (BAPO), an off-policy RLVR framework to improve the data efficiency in large language models post-training. It dynamically selects training batches by re-evaluating historically difficult samples and reusing high-quality ones, while holding a lower bound guarantee for policy improvement. Extensive experiments further demonstrate that BAPO achieves an average 12.5% improvement over GRPO across mathematics, planning, and visual reasoning tasks. Crucially, BAPO successfully resolves 40.7% of problems that base models consistently fail to solve.

1 INTRODUCTION

Reinforcement Learning from Human Feedback (RLHF) has emerged as a transformative paradigm for aligning Large Language Models (LLMs) with human preferences and improving their performance on complex reasoning tasks (Ouyang et al., 2022; Bai et al., 2022). A significant recent evolution is Reinforcement Learning with Verifiable Rewards (RLVR) (Lambert et al., 2024), which replaces costly neural reward models with deterministic verification functions for more efficient and reliable training (Guo et al., 2025). Numerous on-policy RL optimization methods, particularly Group Relative Policy Optimization (GRPO) (Shao et al., 2024), and its variants like Dynamic Sampling Policy Optimization (DAPO) (Yu et al., 2025), Group Sequence Policy Optimization (GSPO) (Zheng et al., 2025), have demonstrated remarkable success in LLM post-training scenarios, achieving exceptional performance on mathematical reasoning, code generation, and various downstream applications (Yang et al., 2025; Chen et al., 2025a; Shen et al., 2025).

Although with lower bound guarantees of policy improvement theoretically (Mroueh, 2025), existing RL post-training frameworks still face significant efficiency challenges in practice. As shown in Figure 1, models after GRPO post-training struggle to handle difficult samples, especially those with zero accuracy in the initial rollout group. The reasons are twofold: (1) **homogeneous rewards**: Investigation from (Hong et al., 2025; Simoni et al., 2025) reveals that both overly difficult and overly simple samples are detrimental to RL post-training, providing minimal policy improvement benefits. This occurs because most RLVR methods' advantage estimation relies heavily on relative reward diversity within each group. When the reward distribution within a rollout group is homogeneous (e.g., entirely identical reward signals), the lower bound guarantee for policy improvement will diverge (Zhang et al.,

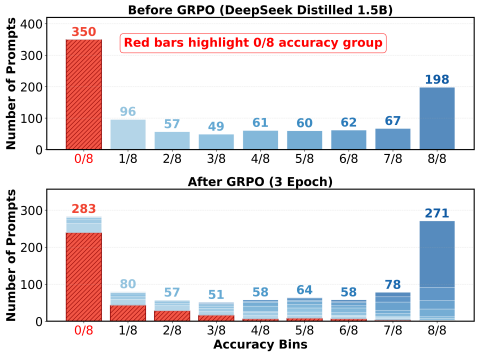

Figure 1: Tracking the sample counts across accuracy groups of the mathematical dataset before and after GRPO post-training.

2025; Mroueh et al., 2025), resulting in minimal effective gradient contribution (Liu et al., 2025; Yu et al., 2025). (2) **waste of experience**: Owing to the sensitivity of policy improvement to intra-group rewards, uneven difficulty distributions within training batches result in substantially fewer actual high-quality samples than the configured batch size. Crucially, being primarily on-policy and lacking experience replay, each rollout group is utilized only once, further wasting valuable training samples (Sun et al., 2025; Li et al., 2025).

A straightforward solution is to adopt off-policy rather than on-policy training paradigms, which has been established in traditional RL tasks as a viable solution to increase sample efficiency and diversity in the training phase (Queeney et al., 2021; Hilton et al., 2022; Meng et al., 2023). However, naively applying sample-reusing schemes to RL frameworks may exacerbate instability during LLM post-training, leading to entropy collapse, and ultimately performance degradation (Yu et al., 2025; He et al., 2025; Chen et al., 2025c).

Thus, to systematically exploring the utility of stale off-policy experience in RLVR post-training, we incorporates multiple off-policy strategies into on-policy RLVR framework to dissect effective pathways for historical data utilization from different dimensions. The main contributions of this paper are as follows:

(1) We propose a difficulty-aware historical experience replay mechanism as a concrete realization of effective off-policy data utilization. Unlike the simple mixing of the buffer's data and online data, we introduce the dynamic selection of high-quality historical samples and online re-evaluation of difficult historical samples to fully harness the power of off-policy experience.

(2) Theoretically, we prove that under certain assumptions, the proposed adaptive construction mechanism can always guarantee that the training batch has a lower bound for policy improvement, thereby laying a theoretical foundation for the framework's stability.

(3) By integrating it into multiple reasoning tasks with different LLM backbones, we validate the proposed **B**atch **A**daptation **P**olicy **O**ptimization (BAPO) method achieves better convergence and yields greater improvements on solving difficult samples compared to existing on-policy and off-policy RLVR frameworks.

2 Related Work

2.1 On-policy RL Post-training Framework

We first review the concept of on-policy RLVR, where the core objective is to optimize an LLM policy to maximize the outcome response reward. Let $x \in \mathcal{X}$ represent the input prompts, and $y \in \mathcal{Y}$ denote responses generated by the LLM policy π_θ. The terminal reward $r(x, y) \in \{0, 1\}$ is determined by a deterministic verification function (Lambert et al., 2024; Guo et al., 2025). Following the setting of GRPO (Shao et al., 2024), the objective is formulated as:

$$\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y_i|}\sum_{t=1}^{|y_i|}\min\left(\rho_{i,t}(\theta)\hat{A}_{i,t}, \text{clip}(\rho_{i,t}(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_{i,t}\right) - \beta \cdot \mathbb{D}_{\text{KL}}(\pi_\theta||\pi_{\text{ref}}) \tag{1}$$

where $\mathcal{G} = \{y_1, y_2, \ldots, y_G\}$ represents a G-size group of responses sampled from $\pi_{\theta_t}(\cdot|x)$ for each input x; $\rho_{i,t}(\theta)$ is the probability ratio $\frac{\pi_\theta(y_i^t|y_i^{<t},x)}{\pi_{\theta_{\text{old}}}(y_i^t|y_i^{<t},x)}$ between current policy and old policy $\pi_{\theta_{\text{old}}}$ for the i-th responses' t-th token, ε limits the magnitude of policy updates; and $\mathbb{D}_{\text{KL}}$ constrains the policy π_θ from deviating too far from a reference policy π_{ref}. Crucially, $\hat{A}_{i,t}$ denotes the estimated advantage of response y_i for input x, which is derived from the standardization of rewards using the statistical properties of group $\mathcal{G}$. For the i-th response $y_i \in \mathcal{G}$ with reward $r_i = r(x, y_i)$, the estimated advantage is:

$$\hat{A}_{i,t} = \frac{r_i - \text{mean}(\{r_\ell\})}{\sqrt{\text{std}^2(\{r_\ell\}) + \varepsilon}} \tag{2}$$

where $\text{mean}(\{r_\ell\})$ and $\text{std}^2(\{r_\ell\})$ are the empirical mean and variance of rewards in group $\mathcal{G}$, respectively.

To enhance the practical efficiency of GRPO, a series of improved on-policy frameworks has been proposed. For instance, DAPO (Yu et al., 2025) sets distinct clipping ranges ε_{low} and $\varepsilon_{\text{high}}$, and employs a dynamic sampling strategy to ensure $\hat{A}_{i,t} \neq 0$. However, it consumes approximately four times the number of rollouts (Qu et al., 2025) compared to GRPO. Meanwhile, GSPO (Zheng et al., 2025) abandons the token-level ratio $\rho_{i,t}(\theta)$ and shifts to the sequence level $s_i(\theta)$, which has been validated to maintain more stable training, particularly in Mixture-of-Experts (MoE) architectures.

While the details of these methods vary, they all adhere to the on-policy framework for sampling and updates: the inference server is updated in synchronization with the trainer parameters, and the sampling strategy follows the " use-once-and-discard" principle throughout the training process.

2.2 Off-policy RL Post-training Framework

In contrast, as shown in Figure 2, off-policy RL post-training frameworks operate under a distinct paradigm, characterized by two core components: off-policy rollout for generating responses and off-policy training for constructing the training batch, as detailed below.

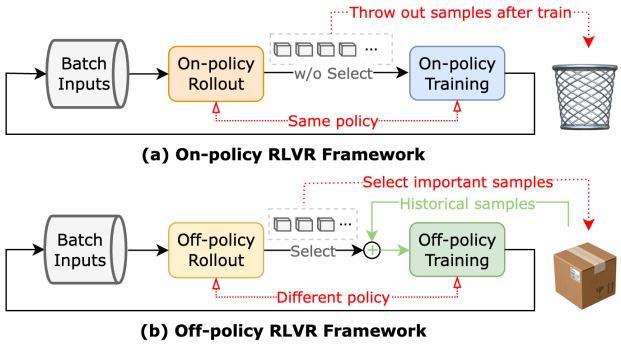

Figure 2: The overview of the (a) on-policy and (b) off-policy RL Post-training framework

Off-policy Rollout avoids exclusive reliance on the current policy for sample generation, instead leveraging past policies or external guidance. For example, AReaL (Fu et al., 2025) employs a fully asynchronous architecture that decouples generation from training, allowing rollout workers to use past policy. Mroueh et al. (Mroueh et al., 2025) also fix the rollout policy on the vLLM inference server for multiple iterations to ensure stable sample generation. LUFFY (Yan et al., 2025) incorporates traces from stronger external policies to enhance reasoning capabilities beyond the model's initial limits. Guide-GRPO (Nath et al., 2025) selectively generates additional rollouts with other guidance when standard rollouts fail.

Off-policy Training uses replay buffers to manage samples from historical policies with varying activation strategies. ARPO (Lu et al., 2025) dynamically samples non-zero reward samples from the buffer only when current batches contain all-zero rewards. DOTS (Sun et al., 2025) maintains a FIFO buffer that consistently reuses recent valid rollouts. RePO (Li et al., 2025) mixes buffer samples with on-policy samples using diverse retrieval strategies. ReMix (Liang et al., 2025) blends samples at fixed ratios while increasing the update-to-data ratio for efficiency. ReLIFT (Ma et al., 2025) stores high-quality solutions to challenging problems in its buffer and refines them through interleaved supervised fine-tuning. Kimi k1.5 (Team et al., 2025) stores both complete and partial trajectories to reduce temporal correlations while maintaining computational efficiency.

However, most off-policy RLVR methods ignore the policy stability of experiences. Samples entering the buffer at different training steps may exhibit varying policy distributions. These discrepancies introduce excessive noise into policy learning, which in turn exacerbates training instability. More importantly, simply reusing historical samples may even hinder the policy's improvement. The high-accuracy historical samples may cause the model to overly focus on existing reasoning paths with high advantages, suppressing the model's exploration capability and resulting in premature convergence to suboptimal solutions (Cui et al., 2025).

3 Method

In this section, we detail the core components of BAPO, particularly the adaptive construction strategy for the training batch, and provide a theoretical guarantee for the training stability of BAPO's policy update. Figure 3 provides an overview of the off-policy rollout and training workflow.

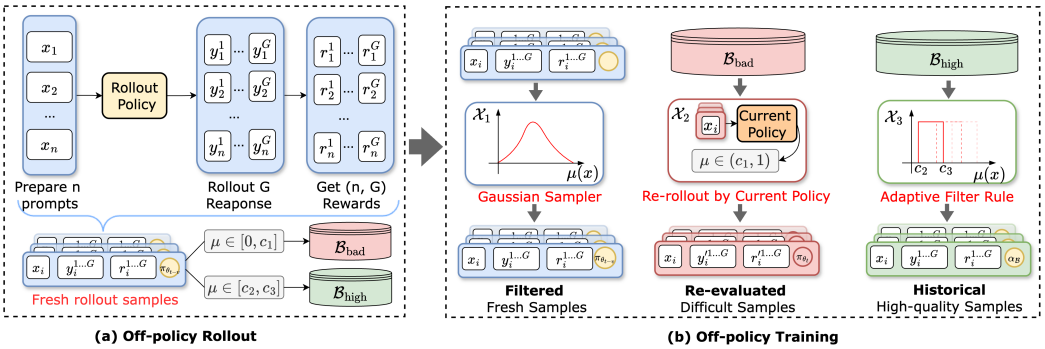

Figure 3: The workflow of (a) off-policy rollout and (b) off-policy training in our RLVR framework

3.1 Formal Definitions

We first formalize our training objective $\mathcal{L}_\alpha(\pi_\theta)$ as a combination of online rollout-derived and historical buffer-derived contributions:

$$\mathcal{L}_\alpha(\pi_\theta) = \underbrace{\mathbb{E}_{(x,y)\sim\alpha}\left[\rho_\alpha(\theta)\cdot\hat{A}(x,y)\right]}_{\text{Contribution from fresh samples}} + \underbrace{\mathbb{E}_{(x,y)\sim\mathcal{B}}\left[\rho_{\alpha_\mathcal{B}}(\theta)\cdot\hat{A}(x,y)\right]}_{\text{Contribution from historical samples}} - \beta\cdot\mathbb{D}_{\text{KL}}(\pi_\theta\|\alpha) \tag{3}$$

where $(x,y)\sim\alpha$ refers to filtered online samples from the delayed rollout policy $\alpha=\pi_{\theta_{t-v}}$ with $v>0$ representing the delay timesteps. $(x,y)\sim\mathcal{B}$ denotes historical samples from the replay buffer $\mathcal{B}$. The importance sampling ratios are defined as $\rho_\alpha=\frac{\pi_\theta(y|x)}{\alpha(y|x)}$ for the online rollout samples and $\rho_{\alpha_\mathcal{B}}=\frac{\pi_\theta(y|x)}{\alpha_\mathcal{B}(y|x)}$ for buffer samples, $\alpha_\mathcal{B}$ is the historical rollout policies that generated the buffer. Each entity in the buffer $\mathcal{B}$ is formally defined as:

$$\mathcal{B}=\{(u_i,\{x_{i,j}\}_{j=1}^G,\{y_{i,j}\}_{j=1}^G,\{r_{i,j}\}_{j=1}^G,\ \{\alpha_\mathcal{B}(y_{i,j}|x_i)\}_{j=1}^G)\}_{i=1}^{|\mathcal{B}|} \tag{4}$$

where u_i is the unique identifier of each prompt, $\{x_{i,j}\}$, $\{y_{i,j}\}$, $\{r_{i,j}\}$ represent the set of prompts, generated responses, and corresponding rewards, respectively. $\{\alpha_\mathcal{B}(y_{i,j}|x_i)\}_{j=1}^G$ is the rollout policy's probability, which is stored for calculating $\rho_{\alpha_\mathcal{B}}(\theta)$ when reusing, and $|\mathcal{B}|$ is the buffer size.

3.2 Adaptive Training Batch Construction

The core of off-policy RLVR lies in how to integrate historical experiences with online samples, to maintain non-homogeneous rewards and an appropriate difficulty distribution in each training step. For BAPO, we introduce **a filter function** $I(x)$ in Definition 3.1 that decomposes the data selection criteria for each training step's batch into three parts.

Definition 3.1 (Training Batch Filtering Function)**.** Define $\mu_{\pi,r}(x)=\mathbb{E}_{y\sim\pi(\cdot|x)}[r(x,y)]$ as the expected reward under policy π for input x. The training batch indicator function $I:\mathcal{X}\to\{0,1\}$ is formulated as:

$$I(x)=\underbrace{1_{\{\frac{1}{G}\le\mu_{\alpha,r}(x)\le\frac{G-1}{G}\}}}_{\text{Filtered Fresh}}+\underbrace{1_{\{\mu_{\alpha_\mathcal{B},r}(x)\le c_1\wedge\mu_{\pi_{\theta_t},r}(x)>c_1\}}}_{\text{Improved Historical Difficult}}+\underbrace{1_{\{c_2\le\mu_{\alpha_\mathcal{B},r}(x)\le c_3\}}}_{\text{Historical High-quality}} \tag{5}$$

where α denotes the delayed rollout policy and $\alpha_\mathcal{B}$ denotes the policy associated with buffer samples. The function selects samples based on three criteria, yielding subsets $\mathcal{X}_1$, $\mathcal{X}_2$ and $\mathcal{X}_3$ respectively.

Next, we explain the selection principles for $I(x)$ and derive three categories of samples, namely $\mathcal{X}_1$, $\mathcal{X}_2$, and $\mathcal{X}_3$, which are obtained from these three conditions, respectively.

(1) Filtered Fresh Samples ($\mathcal{X}_1$). To prevent gradient vanishing and maintain training stability, we filter the online rollout batch to exclude samples with zero variance (i.e., all correct or all incorrect responses). Specifically, we retain fresh samples where the group mean reward satisfies $\mu_{\alpha,r}(x)\in$

$[\frac{1}{G}, \frac{G-1}{G}]$. While other filtering strategies (e.g., Gaussian sampling or uniform sampling) can be applied, we find that simple truncation sufficient for effective learning. A detailed discussion and comparison of different online filtering functions are provided in Appendix A.3.

(2) Improved Historical Difficult Samples ($\mathcal{X}_2$). Samples exhibiting extremely low group mean rewards, where $\mu_{\alpha,r}(x) \in [0, c_1]$, present significant challenges to the current policy and typically yield negligible training gradients. However, as the model evolves, these historically difficult queries may eventually become solvable, transforming into high-value samples for policy improvement. To harness this, we *periodically* re-generate responses using the current policy π_{θ_t} every m training steps and construct the subset $\mathcal{X}_2$ based on the observable improvement.

Let $\mathcal{B}_{\text{bad}} \subseteq \mathcal{B}$ denote the buffer for difficult samples. To manage the computational overhead associated with the re-evaluation process, we limit the buffer capacity $|\mathcal{B}|$ to be equal to the training batch size. A First-In-First-Out (FIFO) mechanism is employed to automatically discard outdated samples when the buffer reaches capacity. $\mathcal{X}_2$ is formulated as:

$$\mathcal{X}_2 = \left\{(x, y') \mid (x, y) \in \mathcal{B}_{\text{bad}}, y' \sim \pi_{\theta_t}(\cdot \mid x), c_1 < \mu_{\pi_{\theta_t},r}(x) < 1\right\} \tag{6}$$

where y' represents the new response generated by π_{θ_t}, and we specifically select samples that show improvement such that $c_1 < \mu_{\pi_{\theta_t},r}(x) < 1$.

(3) Reused Historical High-quality Samples ($\mathcal{X}_3$). Since the re-evaluation of difficult samples ($\mathcal{X}_2$) occurs intermittently, and the filtered fresh samples ($\mathcal{X}_1$) may not suffice to fill the training batch, particularly in the early stages of training, we maintain a separate buffer $\mathcal{B}_{\text{high}} \subseteq \mathcal{B}$ for storing the high-quality historical samples with $\mu_{\alpha,r}(x) \in [c_2, c_3]$.

Unlike the replayed difficult queries, these historical samples are reused directly. Therefore, to mitigate training instability caused by excessive distributional shift between the historical rollout policy and the current policy, we set the capacity of $\mathcal{B}_{\text{high}}$ to $|\mathcal{B}|$ aligned with the training batch size, and also utilize a FIFO eviction strategy to maintain policy relevance. The subset $\mathcal{X}_3$ is defined as:

$$\mathcal{X}_3 = \{\mathcal{S}\left(\mathcal{B}_{\text{high}}, \max\left(0, |\mathcal{X}| - |\mathcal{X}_1| - |\mathcal{X}_2|\right)\right)\} \tag{7}$$

where $\mathcal{S}(\mathcal{B}_{\text{high}}, k)$ represents k elements randomly sampled from $\mathcal{B}_{\text{high}}$, ensuring batch size for stable distributed training.

As the model's capabilities improve, the definition of a "high-quality" sample effectively shifts towards more difficult instances. To adapt to this, we employ a linear mapping function to dynamically adjust the acceptance interval (c_2, c_3) based on the global average performance r_{tot} of the buffer:

$$c_2 = r_{tot} \cdot (c_2^{\text{high}} - c_2^{\text{low}}) + c_2^{\text{low}} \tag{8}$$

$$c_3 = r_{tot} \cdot (c_3^{\text{high}} - c_3^{\text{low}}) + c_3^{\text{low}} \tag{9}$$

where c^{high} and c^{low} are hyperparameters. This mechanism ensures that as r_{tot} increases, the interval (c_2, c_3) shifts to capture harder samples, thereby sustaining the learning challenge.

3.3 Theoretical Analysis

In this section, we further provide theoretical analysis in Theorem 3.2 to establish BAPO's training stability based on (Mroueh et al., 2025)'s theorem. We show that, under certain assumptions, our constructed adaptive batches can consistently maintain guaranteed bounded policy improvement.

Theorem 3.2 (Policy Improvement Lower Bound with Adaptive Training Batch). *Assume rewards are bounded: $0 \leq r \leq 1$. Let π_{θ_t} be the current policy, $\alpha_1 = \pi_{\theta_{t-v}}$ be the delayed rollout policy, $\alpha_2 = \pi_{\theta_t}$ be the current policy for re-evaluation, $\alpha_3 = \alpha_{\mathcal{B}}$ be the buffer policy distribution, and $I(x)$ be the filtering function partitioning samples into $\mathcal{X}_1$, $\mathcal{X}_2$, and $\mathcal{X}_3$.*

Suppose $c_1, c_2, c_3 \in (0, 1)$ with $c_2 < c_3$, and the following TV distance constraints hold:

$$TV(\pi_{\theta_t}(\cdot|x), \pi_{\theta_{t-v}}(\cdot|x)) \leq \delta_1 \quad \forall x \in \mathcal{X}_1 \tag{10}$$

$$TV(\pi_{\theta_t}(\cdot|x), \alpha_{\mathcal{B}}(\cdot|x)) \leq \delta_3 \quad \forall x \in \mathcal{X}_3 \tag{11}$$

where $\delta_1, \delta_3 > 0$ are sufficiently small such that the variance lower bounds remain positive.

Then, for the policy update objective in Equation 3, the expected policy improvement over filtered samples satisfies:

$$\mathbb{E}_{x\sim\rho_{\mathcal{X}}}[I(x)(J(\pi_\theta(\cdot|x)) - J(\pi_{\theta_t}(\cdot|x)))] \geq \sum_{i=1}^{3} \mathcal{L}_i(\pi_\theta, \alpha_i)$$

where:

$$J\big(\pi_\theta(\cdot \mid x)\big) \;=\; \mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\, r(x,y)$$

$$\mathcal{L}_i(\pi_\theta, \alpha_i) = \mathbb{E}_{x\in\mathcal{X}_i}\left[L_{\alpha_i}(\pi_\theta(\cdot|x)) - 2K_i \cdot TV(\pi_\theta(\cdot|x), \alpha_i(\cdot|x)) - 2TV(\pi_{\theta_t}(\cdot|x), \alpha_i(\cdot|x))\right]$$

with $L_{\alpha_i}(\pi_\theta(\cdot|x)) = \frac{1}{\sigma_{\alpha_i,r,\varepsilon}(x)}(J(\pi_\theta(\cdot|x)) - J(\alpha_i(\cdot|x)))$. *The constants are:*

$$K_1 = \frac{1 - \sqrt{\frac{G-1}{G^2} + \varepsilon}}{\sqrt{\frac{G-1}{G^2} + \varepsilon}} \tag{12}$$

$$K_2 = \frac{1 - \sqrt{c_1(1-c_1) + \varepsilon}}{\sqrt{c_1(1-c_1) + \varepsilon}} \tag{13}$$

$$K_3 = \frac{1 - \sqrt{\min(c_2(1-c_2), c_3(1-c_3)) + \varepsilon}}{\sqrt{\min(c_2(1-c_2), c_3(1-c_3)) + \varepsilon}} \tag{14}$$

More importantly, we highlight several properties from this theorem:

Bounded Stability. All constants K_1, K_2, and K_3 are finite positive values, which guarantee that the training process remains numerically stable and theoretically bounded.

Flexible Tightness. The dependence of K_3 on c_2 and c_3 directly reflects our difficulty-aware strategy. When the interval $[c_2, c_3]$ is closer to the boundary values (0 or 1), the resulting bounds become looser but can include more diverse samples.

Off-policy Tolerance. The stability of trust-region methods inherently constrain the magnitude of single-step policy updates. Consequently, the divergence between the current policy π_{θ_t} and the delayed rollout policy α remains bounded over short intervals. Furthermore, the strict FIFO mechanism with limited buffer capacity ensures that only samples from recent policies are retained, thereby maintaining policy consistency within the training batch.

4 Experimental Setup

To comprehensively evaluate the effectiveness of our off-policy RLVR framework, we conduct extensive experiments across different tasks and backbones, following the experimental setup described in (Qu et al., 2025).

First, we select three representative reasoning tasks, as detailed below:

Mathematics. Following prior work (Luo et al., 2025), we use the DeepSeek R1 Distilled 1.5B (Guo et al., 2025) as the base model, and conducted post-training on the DeepScaleR-Preview-Dataset (Aggarwal & Welleck, 2025), which contains 40 thousand question-answer pairs sourced from several mathematics competitions. Evaluation is performed on multiple mathematics benchmarks, including AIME24, AMC23, MATH500 (Hendrycks et al., 2021), Minerva Math (Minerva) (Lewkowycz et al., 2022), and OlympiadBench (Olympiad)He et al. (2024).

Planning. We choose Qwen2.5 Math 1.5B and 7B (Yang et al., 2024) as the backbone, and adopted the Countdown Number Game as the specific task. For training, we used a 10,000-problem subset of the Countdown-34 dataset, where each problem provides 3-4 source numbers. Evaluation was conducted on two variants: Countdown-3to4 (CD-34) test set using a 200-problem held-out split, and the more challenging Countdown-4 (CD-4) test set with 200 problems that consistently provide four source numbers (Chen et al., 2025b).

Visual Geometry. We train Qwen2.5 VL 3B and 7B (Bai et al., 2025) on the 2,101-problem training split of the Geometry3K dataset (Lu et al., 2021), where each problem consists of a geometric

diagram paired with a natural language question requiring spatial and logical reasoning. Evaluation was performed on the official 300-problem validation split (Geo-3K val) and 601-problem test split of Geometry3K (Geo-3K test).

Besides, we select several on-policy and off-policy RLVR frameworks as baselines:

On-policy. We select GRPO (Shao et al., 2024), DAPO (Yu et al., 2025), and MoPPS (Qu et al., 2025) as representative on-policy RLVR methods. GRPO is the first to integrate group-relative advantage estimation into the RLVR framework, while DAPO further improves training stability and efficiency. MoPPS incorporates difficulty-aware prediction into prompt selection.

Off-policy. We compare our approach with three representative off-policy methods: GRPO ($v = 5$) (Mroueh et al., 2025), RePO (Li et al., 2025), and Remix-GRPO (Liang et al., 2025). Specifically, GRPO ($v = 5$) delays the rollout policy with a frequency of 5, whereas RePO and Remix-GRPO adopt diverse replay strategies to retrieve off-policy samples from a replay buffer.

Implementation Details. All comparative experiments were run on 8 A100 GPUs with 80GB memory based on the Verl framework (Sheng et al., 2025). Identical parameters were used to ensure fair comparison, with specific details in Appendix A.7.

5 Results Analysis

5.1 Main Results

We evaluate BAPO across three reasoning tasks to demonstrate its broad applicability. Experimental results show that BAPO consistently outperforms existing baselines throughout training (Figure 4) and testing (Figure 11). Notably, in mathematical tasks, the GRPO baseline exhibits severe training instability, as evidenced by significant oscillations in its early-stage training curve. This is attributed to the high variance in problem difficulty within the DeepScalerR dataset. Under the same settings, BAPO achieves smoother convergence and higher reward bounds.

In test benchmarks (Tables 1), BAPO achieves an **average 12.5% accuracy improvement** over baselines. Crucially, while DAPO approaches BAPO's performance in some metrics, it requires approximately **2.5× more rollouts** (as visualized in Figure 12), imposing a substantial computational burden.

Takeaway 1: Efficiency & Stability

BAPO significantly outperforms on-policy baselines GRPO in convergence stability and exceeds heavy-sampling methods DAPO with significantly fewer rollout costs.

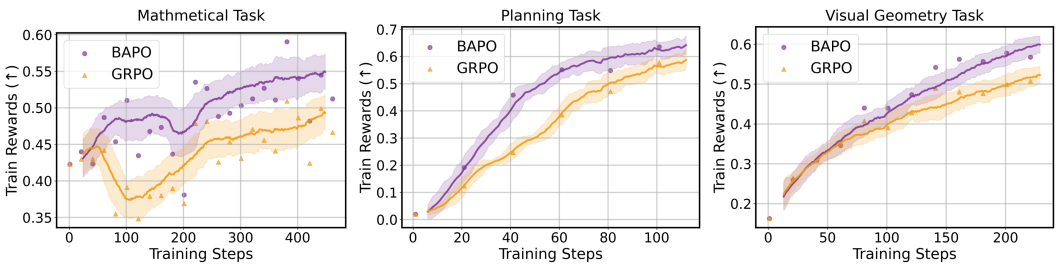

Figure 4: **Training Curves of Reward Changes** for mathematics, planning, and geometry tasks using DeepSeek Distilled Qwen 1.5B, Qwen2.5 Math 1.5B, and Qwen2.5 VL 3B, respectively.

5.2 Mechanism Analysis

To deeply investigate whether BAPO's success stems from sensitive hyperparameter tuning or its core batch reconstruction mechanism, we conducted both **Minimalist Verification** and **Hyperparameter Robustness** experiments.

Table 1: **Comprehensive Evaluation Results.** '+' indicates fine-tuning via the corresponding method. Accuracy is averaged over 32 runs. The **bold** value denotes the top result, and the underlined value denotes the second-top result.

(a) Mathematics Benchmarks

Method	AIME24	AMC	MATH500	Minerva.	Olympiad.	Avg. ↑	Rollouts ↓	Type
DeepSeek R1 Distill Qwen 1.5B	28.80	62.90	82.80	26.50	44.42	48.90	-	-
+GRPO (Guo et al., 2025)	30.73	67.47	85.40	28.95	45.33	51.58	**677k**	on
+DAPO (Yu et al., 2025)	35.73	70.08	86.05	**30.70**	48.48	54.20	1921k	on
+MoPPS* (Qu et al., 2025)	33.33	65.29	84.94	28.88	45.93	51.67	737k	on
+GRPO ($v = 5$) (Mroueh et al., 2025)	30.49	65.09	86.72	28.16	46.18	51.57	**677k**	off
+RePO (Li et al., 2025)	30.42	64.76	83.75	28.33	45.44	50.54	**677k**	off
+Remix-GRPO* (Liang et al., 2025)	33.33	65.06	84.60	26.10	43.55	50.53	-	off
+BAPO (Ours)	**38.54**	**72.74**	**89.18**	29.55	**50.06**	**56.01**	733k	off

(b) Planning and Visual Geometry Benchmarks

Method	CD-34	CD-4	Avg	Method	Geo-3K(val)	Geo-3K(test)	Avg
Qwen2.5 Math 1.5B	1.12	0.37	0.75	Qwen2.5 VL 3B	14.77	19.18	16.98
+GRPO (Guo et al., 2025)	62.94	35.88	49.41	+GRPO (Guo et al., 2025)	36.44	43.12	39.78
+DAPO (Yu et al., 2025)	70.56	45.87	58.22	+DAPO (Yu et al., 2025)	**40.11**	45.18	42.65
+BAPO w/o $\mathcal{X}_2$ (Ours)	60.31	35.31	47.81	+BAPO w/o $\mathcal{X}_2$ (Ours)	30.57	36.92	33.75
+BAPO w/o $\mathcal{X}_3$ (Ours)	64.43	38.75	51.59	+BAPO w/o $\mathcal{X}_3$ (Ours)	32.22	39.79	36.01
+BAPO (Ours)	**73.00**	**47.50**	**60.25**	+BAPO (Ours)	**40.11**	**46.33**	**43.22**
Qwen2.5 Math 7B	2.68	0.94	1.81	Qwen2.5 VL 7B	30.40	36.10	33.25
+GRPO (Guo et al., 2025)	70.75	50.25	60.50	+GRPO (Guo et al., 2025)	40.79	47.15	43.97
+DAPO (Yu et al., 2025)	78.75	**57.43**	68.09	+DAPO (Yu et al., 2025)	40.87	47.02	43.95
+BAPO (Ours)	**79.13**	57.13	**68.13**	+BAPO (Ours)	**41.89**	**48.77**	**45.33**

*This method's performance is taken from the corresponding paper.

Takeaway 2: Off-policy Components > Off-policy Hyperparameters

The performance gains of BAPO primarily stem from the structural logic of its off-policy components rather than specific hyperparameter settings. The framework remains effective even under rigid, parameter-free conditions.

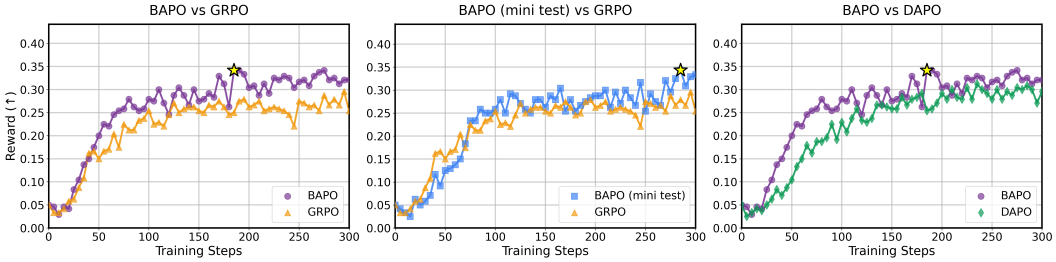

Figure 5: **Test Curves of Group Accuracy Changes** on AIME for different RLVR methods based on **Qwen3 8B**. Left: Standard BAPO vs. GRPO. Medium: **BAPO (mini test) vs. GRPO**. Right: Standard BAPO vs. DAPO. Even with fixed, theoretically-derived selection criteria, BAPO (mini test) yields higher rewards than GRPO.

Minimalist Verification. To validate the theoretical implications of Theorem 3.2 without relying on hyperparameter engineering, specifically avoiding the tuning of thresholds c_1, c_2, c_3 and update frequencies, we devised a **"Mini-test"** experiment. We trained Qwen3 8B on the mathematics task under 4K length constraints using a stripped-down, parameter-free BAPO logic for constructing training batch:

$\mathcal{X}_1$**:** We apply strictly standard zero-advantage filtering, removing only the prompts where all G responses are entirely correct or entirely wrong. This ensures non-vanishing gradients without additional selection criteria.

$\mathcal{X}_2$**:** We replay exclusively historical *all-wrong* samples ($\mu_{\alpha,r}(x) = 0$). These correspond exactly to the difficult cases discarded by $\mathcal{X}_1$, creating a closed-loop system that recovers waste data without requiring a "difficulty threshold".

$\mathcal{X}_3$**:** Instead of a dynamic accuracy range, we reuse historical samples with exactly **50% accuracy**. As formally proven in Proposition A.3, samples with accuracy $\mu_{\alpha,r}(x) = \frac{1}{2}$ maximize the reward variance, thereby providing the theoretical maximum potential for single-step policy improvement $J(\pi_\theta) - J(\pi_{\theta_t})$.

The results in Figure 5 (Right) demonstrate that even in the hyperparameter-free "Mini-test", BAPO maintains a clear advantage over GRPO. This confirms that the structural introduction of $\mathcal{X}_2$ and $\mathcal{X}_3$ drives the performance, not the specific tuning of c values.

Component Efficacy. To evaluate the contribution of $\mathcal{X}_2$ (re-evaluated difficult samples) and $\mathcal{X}_3$ (reused high-quality samples), we conduct ablation studies shown in Table 1. Both components are essential: removing $\mathcal{X}_2$ causes a $\sim$21% performance drop, underscoring the importance of explicitly targeting difficult samples.

Hyperparameter Robustness. We further evaluate the sensitivity of BAPO to its key hyperparameters: rollout delay v, re-rollout frequency m, and difficulty thresholds. **Frequency** (v, m)**:** As shown in Figure 6 (Column 1), performance remains stable within reasonable ranges (e.g., $v = 5, m = 5$). Extreme delays only degrade performance when policy divergence becomes excessive, aligning with our theoretical analysis regarding the trust region. **Difficulty Thresholds** (c_2, c_3)**:** While our adaptive boundary mechanism yields the best convergence, Figure 6 (Column 3) shows that using fixed ranges still significantly outperforms baselines. This indicates that the *presence* of diverse historical data is more critical than the precise values of the thresholds.

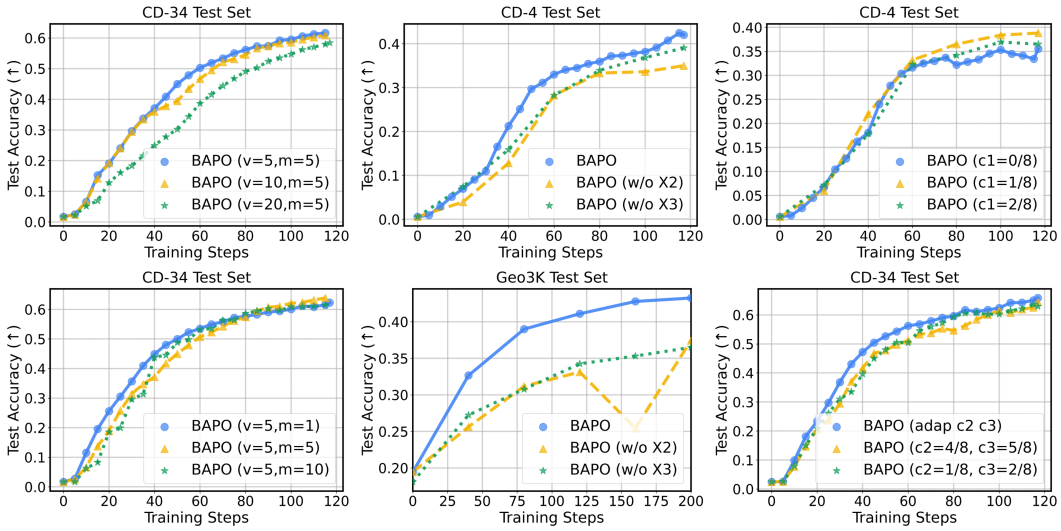

Figure 6: **Ablation Studies** for BAPO. The first column presents ablations on frequency-related hyperparameters (m, v). The second column shows ablations on buffer subsets $(\mathcal{X}_2, \mathcal{X}_3)$. The third column compares fixed vs. adaptive difficulty thresholds.

5.3 Detailed Analysis

We analyze BAPO's internal mechanisms below. For extended analysis on training dynamics, computation, and visualization, please refer to **Appendices A.4, A.5 and A.6**.

Tracking Difficult Samples. We visualize the training dynamics in Figure 7. BAPO exhibits a superior capability to "unlock" difficult problems: after 3 epochs, BAPO successfully improves **31%** of the samples that were initially unsolvable ($0/8$ accuracy), compared to only **19%** for GRPO.

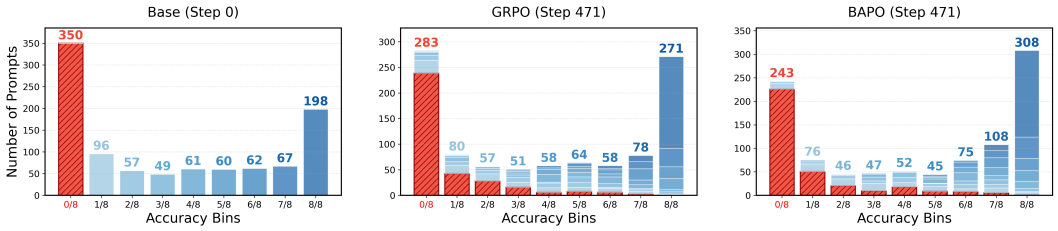

Figure 7: Tracking changes in the **Number of Different Accuracy Bins** on the DeepScalerR training subset. Special attention is paid to the reduction of bad samples (red bars).

Sample Distribution & Efficiency. To uncover the source of BAPO's efficiency, we analyze the dynamic batch construction in Figure 8 alongside the rollout costs in Figure 12.

As observed in Figure 8, the assembled training batch size frequently fluctuates below the maximum configured capacity. This reduction in backward propagation load effectively offsets the computational overhead caused by off-policy re-evaluation and log-probability re-computation. Consequently, as detailed in Table. 2, BAPO maintains a training speed comparable to GRPO while requiring significantly fewer rollouts than DAPO, achieving a superior trade-off between convergence performance and computational cost.

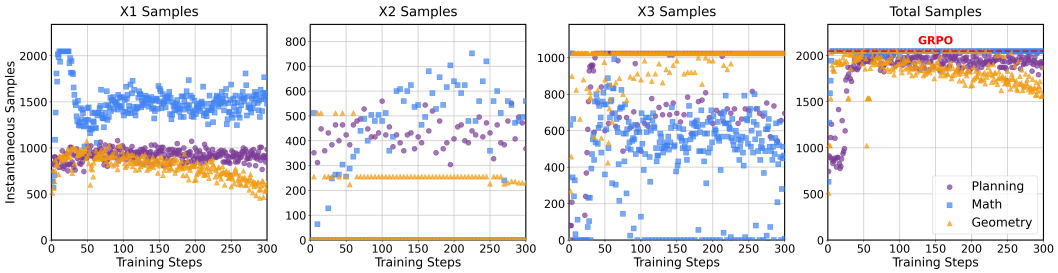

Figure 8: **Dynamic Sample Distribution.** The composition of BAPO's $\mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3$ and the total samples compared to the fixed GRPO batch size (Red line).

Takeaway3: Efficient Batch Adaptation

BAPO optimizes efficiency by dynamically balancing fresh and historical data. By often training on effective batches smaller than the maximum capacity, it offsets replay overheads, achieving superior performance with training speeds comparable to the lightweight GRPO baseline.

6 Conclusion

In this paper, we propose BAPO, an off-policy RL framework for LLM post-training. It aims to utilize historical training data better and thereby improve training efficiency. Specifically, we appropriately delay the rollout policy to stabilize the policy discrepancies of buffer samples. More importantly, we construct training batches by re-evaluating difficult samples and reusing historical high-quality ones, thereby enhancing the efficiency of post-training. We validate the strong adaptability of the BAPO framework through experiments on three distinct reasoning tasks using different LLM backbones, and the results demonstrate that BAPO significantly outperforms baselines in both convergence performance and training efficiency. Nevertheless, exploring how to adapt BAPO to large models with MoE architectures, as well as how to tailor BAPO to agent-based RL frameworks, remains a significant challenge.

Ethics Statement

All authors of this study strictly adhere to the ICLR code of ethics. Our research does not involve any potential conflicts of interest or sponsorship issues. We have carefully considered and addressed concerns related to discrimination, bias, and fairness in our methodology. The study raises no privacy or security concerns, maintains full legal compliance, and upholds the highest standards of research integrity. All experimental procedures and data handling practices follow established ethical guidelines for machine learning research.

Reproducibility statement

To ensure full reproducibility of our results, we provide comprehensive implementation details of the proposed BAPO training algorithm in the supplementary materials. All experimental settings, hyperparameters, and dataset specifications are clearly documented. For our theoretical contributions, complete proofs and clear explanations of all assumptions are included in the appendix. Code and data will be made available upon acceptance to facilitate replication of our findings.

The Use of Large Language Models

In this research, we employed LLMs solely as language editing tools to improve the clarity and readability of our manuscript. LLMs were used for grammar checking, style refinement, and language polishing purposes only. All core research ideas, experimental design, analysis, and conclusions are entirely the original work of the authors. The use of LLMs did not contribute to the conceptual or technical content of this study.

References

Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning. *arXiv preprint arXiv:2503.04697*, 2025.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *arXiv preprint arXiv:2503.09567*, 2025a.

Xiaoyin Chen, Jiarui Lu, Minsu Kim, Dinghuai Zhang, Jian Tang, Alexandre Piché, Nicolas Gontier, Yoshua Bengio, and Ehsan Kamalloo. Self-evolving curriculum for llm reasoning. *arXiv preprint arXiv:2505.14970*, 2025b.

Yang Chen, Zhuolin Yang, Zihan Liu, Chankyu Lee, Peng Xu, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Acereason-nemotron: Advancing math and code reasoning through reinforcement learning. *arXiv preprint arXiv:2505.16400*, 2025c.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, et al. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025.

Wei Fu, Jiaxuan Gao, Xujie Shen, Chen Zhu, Zhiyu Mei, Chuyi He, Shusheng Xu, Guo Wei, Jun Mei, Jiashu Wang, et al. Areal: A large-scale asynchronous reinforcement learning system for language reasoning. *arXiv preprint arXiv:2505.24298*, 2025.

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

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*, 2024.

Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, et al. Skywork open reasoner 1 technical report. *arXiv preprint arXiv:2505.22312*, 2025.

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

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

Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, et al. Glm-4.1 v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*, 2025.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

Siheng Li, Zhanhui Zhou, Wai Lam, Chao Yang, and Chaochao Lu. Repo: Replay-enhanced policy optimization. *arXiv preprint arXiv:2506.09340*, 2025.

Jing Liang, Hongyao Tang, Yi Ma, Jinyi Liu, Yan Zheng, Shuyue Hu, Lei Bai, and Jianye Hao. Squeeze the soaked sponge: Efficient off-policy reinforcement finetuning for large language model. *arXiv preprint arXiv:2507.06892*, 2025.

Ziru Liu, Cheng Gong, Xinyu Fu, Yaofang Liu, Ran Chen, Shoubo Hu, Suiyun Zhang, Rui Liu, Qingfu Zhang, and Dandan Tu. Ghpo: Adaptive guidance for stable and efficient llm reinforcement learning. *arXiv preprint arXiv:2507.10628*, 2025.

Fanbin Lu, Zhisheng Zhong, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Arpo: End-to-end policy optimization for gui agents with experience replay. *arXiv preprint arXiv:2505.16282*, 2025.

Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-chun Zhu. Inter-gps: Interpretable geometry problem solving with formal language and symbolic reasoning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 6774–6786, 2021.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Li Erran Li, et al. Deepscaler: Surpassing o1-preview with a 1.5 b model by scaling rl. *Notion Blog*, 2025.

Lu Ma, Hao Liang, Meiyi Qiang, Lexiang Tang, Xiaochen Ma, Zhen Hao Wong, Junbo Niu, Chengyu Shen, Runming He, Bin Cui, et al. Learning what reinforcement learning can't: Interleaved online fine-tuning for hardest questions. *arXiv preprint arXiv:2506.07527*, 2025.

Wenjia Meng, Qian Zheng, Gang Pan, and Yilong Yin. Off-policy proximal policy optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 9162–9170, 2023.

Youssef Mroueh. Reinforcement learning with verifiable rewards: Grpo's effective loss, dynamics, and success amplification. *arXiv preprint arXiv:2503.06639*, 2025.

Youssef Mroueh, Nicolas Dupuis, Brian Belgodere, Apoorva Nitsure, Mattia Rigotti, Kristjan Greenewald, Jiri Navratil, Jerret Ross, and Jesus Rios. Revisiting group relative policy optimization: Insights into on-policy and off-policy training. *arXiv preprint arXiv:2505.22257*, 2025.

Vaskar Nath, Elaine Lau, Anisha Gunjal, Manasi Sharma, Nikhil Baharte, and Sean Hendryx. Adaptive guidance accelerates reinforcement learning of reasoning models. *arXiv preprint arXiv:2506.13923*, 2025.

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

Yun Qu, Qi Cheems Wang, Yixiu Mao, Vincent Tao Hu, and Xiangyang Ji. Can prompt difficulty be online predicted for accelerating rl finetuning of reasoning models? *arXiv preprint arXiv:2507.04632*, 2025.

James Queeney, Yannis Paschalidis, and Christos G Cassandras. Generalized proximal policy optimization with sample reuse. *Advances in Neural Information Processing Systems*, 34:11909–11919, 2021.

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

Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, et al. Vlm-r1: A stable and generalizable r1-style large vision-language model. *arXiv preprint arXiv:2504.07615*, 2025.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.

Marco Simoni, Aleksandar Fontana, Giulio Rossolini, and Andrea Saracino. Gtpo: Trajectory-based policy optimization in large language models, 2025. URL https://arxiv.org/abs/2508.03772.

Yifan Sun, Jingyan Shen, Yibin Wang, Tianyu Chen, Zhendong Wang, Mingyuan Zhou, and Huan Zhang. Improving data efficiency for llm reinforcement fine-tuning through difficulty-targeted online data selection and rollout replay. *arXiv preprint arXiv:2506.05316*, 2025.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to reason under off-policy guidance. *arXiv preprint arXiv:2504.14945*, 2025.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

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

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

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

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

A Appendix

A.1 Glossary of Terms and Notations

Term	Definition
c_1, c_2, c_3	Thresholds for classifying historical samples by difficulty (group mean reward).
$\mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3$	Subsets of training batch: fresh, re-evaluated difficult, and historical high-quality samples.
m	Re-evaluation frequency for historically difficult samples.
v	Delay steps for updating the rollout policy.
G	Group size, number of responses generated per prompt during rollout.
$\mathcal{B}$	Replay buffer storing historical samples.
$\hat{A}_{i,t}$	Estimated advantage for token t in response i.
ε	Clipping parameter in PPO-style objectives.
β	Coefficient for KL penalty in the objective function.
$I(x)$	Filter function for constructing BAPO's training batch.
$\mathbb{D}_{\text{KL}}$	Kullback–Leibler divergence, used to constrain policy deviation.
α	Rollout policy for BAPO, which synchronizes to π_θ every v steps.
π_θ	LLM policy parameterized by θ.
π_{ref}	Reference policy (e.g., initial pre-trained model).
$\rho(\theta)$	Importance sampling ratio: $\frac{\pi_\theta(y\|x)}{\pi_{\text{old}}(y\|x)}$.
$r(x, y)$	Reward function, we set to binary (0/1) based on correctness.
$\mu_{\alpha,r}(x)$	Expected reward under policy π for input x. We approximate this value using the mean of $r(x, y)$ corresponding to G responses y generated by the rollout policy α for each prompt x.
$\sigma_{\alpha,r,\varepsilon}(x)$	Standard deviation of rewards under policy α for input x, with smoothing ε.
$J(\pi(\cdot\|x))$	Expected reward of policy π for input x: $\mathbb{E}_{y\sim\pi(\cdot\|x)}[r(x, y)]$.
$\mathcal{N}(\mu_{\alpha,r}(x) \mid \mu, \sigma^2)$	A sampling method that assigns weights to online rollouts based on a normal distribution centered at μ with standard deviation σ, used to filter samples by their group mean reward $\mu_{\alpha,r}(x)$.

A.2 Theoretical Analysis

Lemma A.1 (Kantorovich-Rubenstein duality of total variation distance)**.** *The Kantorovich-Rubinstein duality (variational representation) of the total variation distance is as follows:*

$$TV(m_1, m_2) = \frac{1}{2L} \sup_{g\in\mathcal{G}_L} \left\{\mathbb{E}_{Z\sim m_1}[g(Z)] - \mathbb{E}_{Z\sim m_2}[g(Z)]\right\}, \tag{15}$$

where $\mathcal{G}_L = \{g : \mathcal{Z} \to \mathbb{R}, \|g\|_\infty \le L\}$.

Theorem A.2 (Policy Improvement Lower Bound with Adaptive Training Batch). *Assume rewards are bounded:* $0 \le r \le 1$. *Let* π_{θ_t} *be the current policy,* $\alpha_1 = \pi_{\theta_{t-v}}$ *be the delayed rollout policy,* $\alpha_2 = \pi_{\theta_t}$ *be the current policy for re-evaluation,* $\alpha_3 = \alpha_{\mathcal{B}}$ *be the buffer policy distribution, and* $I(x)$ *be the filtering function partitioning samples into* $\mathcal{X}_1$*,* $\mathcal{X}_2$*, and* $\mathcal{X}_3$*.*

Suppose $c_1, c_2, c_3 \in (0, 1)$ *with* $c_2 < c_3$*. and the following TV distance constraints hold:*

$$TV(\pi_{\theta_t}(\cdot|x), \pi_{\theta_{t-v}}(\cdot|x)) \le \delta_1 \quad \forall x \in \mathcal{X}_1 \tag{16}$$

$$TV(\pi_{\theta_t}(\cdot|x), \alpha_{\mathcal{B}}(\cdot|x)) \le \delta_3 \quad \forall x \in \mathcal{X}_3 \tag{17}$$

where $\delta_1, \delta_3 > 0$ *are sufficiently small such that the variance lower bounds remain positive.*

Then, for the policy update objective in Equation 3, the expected policy improvement over filtered samples satisfies:

$$\mathbb{E}_{x\sim\rho_{\mathcal{X}}}[I(x)(J(\pi_\theta(\cdot|x)) - J(\pi_{\theta_t}(\cdot|x)))] \ge \sum_{i=1}^{3} \mathcal{L}_i(\pi_\theta, \alpha_i)$$

where:

$$\mathcal{L}_i(\pi_\theta, \alpha_i) = \mathbb{E}_{x\in\mathcal{X}_i}\left[L_{\alpha_i}(\pi_\theta(\cdot|x)) - 2K_i \cdot TV(\pi_\theta(\cdot|x), \alpha_i(\cdot|x)) - 2TV(\pi_{\theta_t}(\cdot|x), \alpha_i(\cdot|x))\right]$$

with $L_{\alpha_i}(\pi_\theta(\cdot|x)) = \frac{1}{\sigma_{\alpha_i,r,\varepsilon}(x)}(J(\pi_\theta(\cdot|x)) - J(\alpha_i(\cdot|x)))$. *The constants are:*

$$K_1 = \frac{1 - \sqrt{\frac{G-1}{G^2} + \varepsilon}}{\sqrt{\frac{G-1}{G^2} + \varepsilon}} \tag{18}$$

$$K_2 = \frac{1 - \sqrt{c_1(1-c_1) + \varepsilon}}{\sqrt{c_1(1-c_1) + \varepsilon}} \tag{19}$$

$$K_3 = \frac{1 - \sqrt{\min(c_2(1-c_2), c_3(1-c_3)) + \varepsilon}}{\sqrt{\min(c_2(1-c_2), c_3(1-c_3)) + \varepsilon}} \tag{20}$$

Proof. We prove the bound by analyzing each filtered sample set separately, applying off-policy policy improvement bounds tailored to the reference distribution used in each region.

Step 1: Core inequality for off-policy samples. For any x such that $I(x) = 1$, we establish the fundamental inequality:

$$J(\pi_\theta(\cdot|x)) - J(\pi_{\theta_t}(\cdot|x)) \geq L_{\alpha_i}(\pi_\theta(\cdot|x)) - 2K_i \cdot \text{TV}(\pi_\theta(\cdot|x), \alpha_i(\cdot|x)) \tag{21}$$

$$- 2\text{TV}(\pi_{\theta_t}(\cdot|x), \alpha_i(\cdot|x)) \tag{22}$$

where $K_i = \frac{1-\sigma_{\alpha_i,r,\varepsilon}(x)}{\sigma_{\alpha_i,r,\varepsilon}(x)}$ is a constant that depends on the variance of rewards in each filtered subset.

First, we expand the advantage objective. By definition:

$$L_{\alpha_i}(\pi_\theta(\cdot|x)) = \mathbb{E}_{y\sim\alpha_i(\cdot|x)}\left[\frac{\pi_\theta(y|x)}{\alpha_i(y|x)} A_{\alpha_i}(x,y)\right] \tag{23}$$

$$= \mathbb{E}_{y\sim\alpha_i(\cdot|x)}\left[\frac{\pi_\theta(y|x)}{\alpha_i(y|x)} \cdot \frac{r(x,y) - \mu_{\alpha_i,r}(x)}{\sigma_{\alpha_i,r,\varepsilon}(x)}\right] \tag{24}$$

$$= \frac{1}{\sigma_{\alpha_i,r,\varepsilon}(x)}(J(\pi_\theta(\cdot|x)) - J(\alpha_i(\cdot|x))) \tag{25}$$

Next, we establish the key algebraic identity relating $L_{\alpha_i}(\pi_\theta(\cdot|x))$ to $J(\pi_\theta(\cdot|x)) - J(\pi_{\theta_t}(\cdot|x))$:

$$L_{\alpha_i}(\pi_\theta(\cdot|x)) - (J(\pi_\theta(\cdot|x)) - J(\pi_{\theta_t}(\cdot|x))) \tag{26}$$

$$= \frac{1 - \sigma_{\alpha_i,r,\varepsilon}(x)}{\sigma_{\alpha_i,r,\varepsilon}(x)}(J(\pi_\theta(\cdot|x)) - J(\alpha_i(\cdot|x))) + (J(\pi_{\theta_t}(\cdot|x)) - J(\alpha_i(\cdot|x))) \tag{27}$$

Application of Kantorovich-Rubenstein duality: For bounded rewards with $\|r\|_\infty = 1$, the Kantorovich-Rubenstein duality Lemma A.1 provides:

$$|J(\pi_\theta(\cdot|x)) - J(\alpha_i(\cdot|x))| \leq 2 \cdot \text{TV}(\pi_\theta(\cdot|x), \alpha_i(\cdot|x)) \tag{28}$$

$$|J(\pi_{\theta_t}(\cdot|x)) - J(\alpha_i(\cdot|x))| \leq 2 \cdot \text{TV}(\pi_{\theta_t}(\cdot|x), \alpha_i(\cdot|x)) \tag{29}$$

Since $0 \leq r \leq 1$, we have $\sigma_{\alpha_i,r,\varepsilon}(x) < 1$, ensuring $K_i = \frac{1-\sigma_{\alpha_i,r,\varepsilon}(x)}{\sigma_{\alpha_i,r,\varepsilon}(x)} \geq 0$. Combining these bounds yields the desired inequality.

Step 2: Analysis for $\mathcal{X}_1$ (Filtered fresh samples). For $x \in \mathcal{X}_1$, samples are generated by the delayed rollout policy $\alpha_1 = \pi_{\theta_{t-v}}$ and selected via Gaussian sampling with group-level accuracy $\mu_{\alpha_1,r}(x) \in \{\frac{1}{G}, \frac{2}{G}, \ldots, \frac{G-1}{G}\}$, excluding extremes $\{0, 1\}$.

Variance analysis on discrete set: For the variance function $f(p) = p(1-p)$ over the discrete set $\{\frac{1}{G}, \frac{2}{G}, \ldots, \frac{G-1}{G}\}$, the minimum value occurs at the boundary points $p = \frac{1}{G}$ or $p = \frac{G-1}{G}$, both yielding $f(p) = \frac{G-1}{G^2}$. Therefore:

$$\sigma^2_{\alpha_1,r}(x) = \mu_{\alpha_1,r}(x)(1 - \mu_{\alpha_1,r}(x)) \geq \frac{G-1}{G^2} \tag{30}$$

Thus: $\sigma_{\alpha_1,r,\varepsilon}(x) \geq \sqrt{\frac{G-1}{G^2}+\varepsilon}$, yielding:

$$K_1 = \frac{1-\sqrt{\frac{G-1}{G^2}+\varepsilon}}{\sqrt{\frac{G-1}{G^2}+\varepsilon}}$$

Step 3: Analysis for $\mathcal{X}_2$ (Re-evaluated difficult samples). For $x \in \mathcal{X}_2$, samples are generated by the current policy $\alpha_2 = \pi_{\theta_t}$ through re-evaluation of historically difficult samples. The selection criterion ensures that historically difficult samples ($\mu_{\alpha_B,r}(x) \leq c_1$) now achieve improved performance ($c_1 < \mu_{\pi_{\theta_t},r}(x) < 1$) under the current policy.

Since these samples are directly generated by π_{θ_t}, we have $\alpha_2 = \pi_{\theta_t}$, and the constraint $c_1 < \mu_{\pi_{\theta_t},r}(x) < 1$ provides a natural lower bound, yielding:

$$\sigma^2_{\alpha_2,r}(x) = \mu_{\alpha_2,r}(x)(1-\mu_{\alpha_2,r}(x)) > c_1(1-c_1) \tag{31}$$

Therefore: $\sigma_{\alpha_2,r,\varepsilon}(x) > \sqrt{c_1(1-c_1)+\varepsilon}$, giving us:

$$K_2 = \frac{1-\sqrt{c_1(1-c_1)+\varepsilon}}{\sqrt{c_1(1-c_1)+\varepsilon}}$$

Step 4: Analysis for $\mathcal{X}_3$ (Historical high-quality samples). For $x \in \mathcal{X}_3$, samples are generated by historical buffer policies $\alpha_3 = \alpha_B$ with $\mu_{\alpha_B,r}(x) \in [c_2, c_3]$.

Since $\mu_{\alpha_3,r}(x)(1-\mu_{\alpha_3,r}(x))$ achieves its minimum at the endpoints of the interval $[c_2, c_3]$:

$$\sigma^2_{\alpha_3,r}(x) \geq \min(c_2(1-c_2), c_3(1-c_3)) \tag{32}$$

Therefore: $\sigma_{\alpha_3,r,\varepsilon}(x) \geq \sqrt{\min(c_2(1-c_2), c_3(1-c_3))+\varepsilon}$, yielding:

$$K_3 = \frac{1-\sqrt{\min(c_2(1-c_2), c_3(1-c_3))+\varepsilon}}{\sqrt{\min(c_2(1-c_2), c_3(1-c_3))+\varepsilon}}$$

Step 5: Combining the results. Taking expectations over $x \sim \rho_{\mathcal{X}}$ and applying the indicator function decomposition:

$$\mathbb{E}_{x\sim\rho_{\mathcal{X}}}[I(x)(J(\pi_\theta(\cdot|x)) - J(\pi_{\theta_t}(\cdot|x)))] \tag{33}$$

$$= \sum_{i=1}^{3} \mathbb{E}_{x\sim\rho_{\mathcal{X}}}[\mathbf{1}_{\{x\in\mathcal{X}_i\}}(J(\pi_\theta(\cdot|x)) - J(\pi_{\theta_t}(\cdot|x)))] \tag{34}$$

$$\geq \sum_{i=1}^{3} \mathbb{E}_{x\in\mathcal{X}_i}[L_{\alpha_i}(\pi_\theta(\cdot|x)) - 2K_i \cdot \mathrm{TV}(\pi_\theta(\cdot|x), \alpha_i(\cdot|x)) - 2\mathrm{TV}(\pi_{\theta_t}(\cdot|x), \alpha_i(\cdot|x))] \tag{35}$$

$$= \sum_{i=1}^{3} \mathcal{L}_i(\pi_\theta, \alpha_i) \tag{36}$$

All constants K_1, K_2, K_3 are finite, since denominators are strictly positive by construction and numerators are bounded by 1 under $c_1, c_2, c_3 \in (0,1)$, completing the proof. □

Proposition A.3. *For binary reward tasks where $r(x,y) \in \{0,1\}$, the contribution to the policy improvement lower bound is maximized when the expected group reward of the sample is $\mu = 0.5$.*

Proof. Recalling Theorem 3.2, the lower bound for policy improvement on a specific data distribution involves the constant K, which scales the penalty for policy divergence. The tightness of this bound is governed by the standard deviation of the rewards $\sigma_{\alpha,r}(x)$.

Due to advantage standardization $\hat{A} \propto \frac{1}{\sigma}$, the effective step size in the advantage estimation and consequently the gradient magnitude is proportional to the inverse of the standard deviation. However, in the context of the lower bound analysis in Theorem 3.2, the stability constant K is defined as:

$$K(\mu) = \frac{1 - \sigma(\mu)}{\sigma(\mu)} \tag{37}$$

where a smaller K indicates a tighter bound and thus a larger guaranteed improvement step. For a binary reward function $r \in \{0, 1\}$, the reward distribution follows a Bernoulli distribution with parameter $\mu(x) = \mathbb{E}[r|x]$. The variance is given by:

$$\sigma^2(\mu) = \mu(1 - \mu) \tag{38}$$

To find the μ that maximizes variance, we take the derivative with respect to μ:

$$\frac{d}{d\mu}(\mu - \mu^2) = 1 - 2\mu \tag{39}$$

Setting the derivative to zero:

$$1 - 2\mu = 0 \implies \mu = 0.5 \tag{40}$$

Since the second derivative $\frac{d^2}{d\mu^2} = -2 < 0$, this is a global maximum.

At $\mu = 0.5$, the variance is maximized ($\sigma^2 = 0.25, \sigma = 0.5$). This corresponds to the state of maximum entropy, where the model is most "uncertain" about the outcome. Training on these samples provides the strongest gradient signal for distinguishing between correct and incorrect reasoning paths, effectively maximizing the information gain per step. Conversely, as $\mu \to 0$ or $\mu \to 1$, $\sigma \to 0$, causing the advantage estimates to numerical instability or the gradient signal to vanish. Therefore, selecting samples with $\mu = 0.5$ theoretically offers the most efficient learning signal and the most favorable stability bound.

□

A.3 Online Filter Mechanism Analysis

To investigate the impact of fresh sample selection on training stability and convergence, we conduct an ablation study using Qwen3 8B with a 4K response length limit. We compare three distinct filtering strategies for the online component ($\mathcal{X}_1$):

Mode 1 (Range Filter): It retains samples with group mean rewards $\mu \in [\frac{1}{G}, \frac{G-1}{G}]$. This effectively removes only the zero-advantage samples (all-correct or all-incorrect) that contribute minimal gradients.

Mode 2 (Gaussian Filter): A difficulty-weighted strategy that prioritizes samples with high variance (accuracy near 0.5) using a Gaussian distribution, thereby reducing the proportion of extremely easy or hard samples.

Mode 3 (Uniform Filter): A baseline that randomly selects 60% of the fresh samples regardless of their quality. This ratio was chosen to match the approximate data retention rates of Mode 1 and Mode 2 (approximately 40%–60%) for a fair comparison of data volume.

The Value of Quality over Randomness. As illustrated in Figure 9, the uniform filter mechanism exhibits severe instability, characterized by exploding gradient norms and a complete collapse in performance after 150 steps. Since this strategy blindly includes all-wrong samples (where $\mu = 0$), the model is forced to update based on low-quality, zero-advantage signals. Suppressing the token probabilities of incorrect responses without a corresponding positive signal introduces significant noise and uncertainty, ultimately destabilizing the policy. This failure highlights that the *quality* of the training batch, particularly the exclusion of zero-advantage noise, is crucial.

Convergence Speed and Final Performance. The Gaussian filter demonstrates faster convergence in the early stages. By focusing heavily on samples with the highest variance (accuracy ≈ 0.5), it provides the steepest learning signal initially. However, its final convergence performance is lower

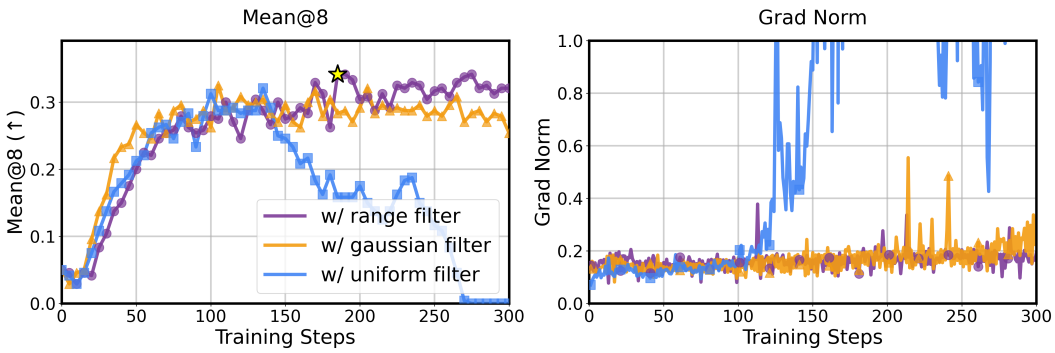

Figure 9: **Ablation on Online Filtering Strategies.** Comparison of Range Filter, Gaussian Filter, and Uniform Filter on training stability (Grad Norm) and performance (Mean@8). The star symbol indicates the best checkpoint for BAPO.

than that of the range filter. We hypothesize that the Gaussian filter restricts sample diversity by aggressively filtering out samples that are slightly easier or harder but still informative. In contrast, the range filter retains a broader spectrum of valid samples. While it learns slightly slower initially, it maintains a rich distribution of training data, preventing premature plateauing and ultimately achieving the highest asymptotic performance.

A.4 Training Dynamics and Test Curves

As illustrated in Figure 10 and Figure 11, we present more detailed training dynamics and test curves for the Planning and Vision Geometry tasks. The results indicate that both BAPO and DAPO consistently outperform GRPO in terms of training rewards. Interestingly, BAPO exhibits higher entropy, reflecting better exploration capability compared to other algorithms, which also results in longer response lengths.

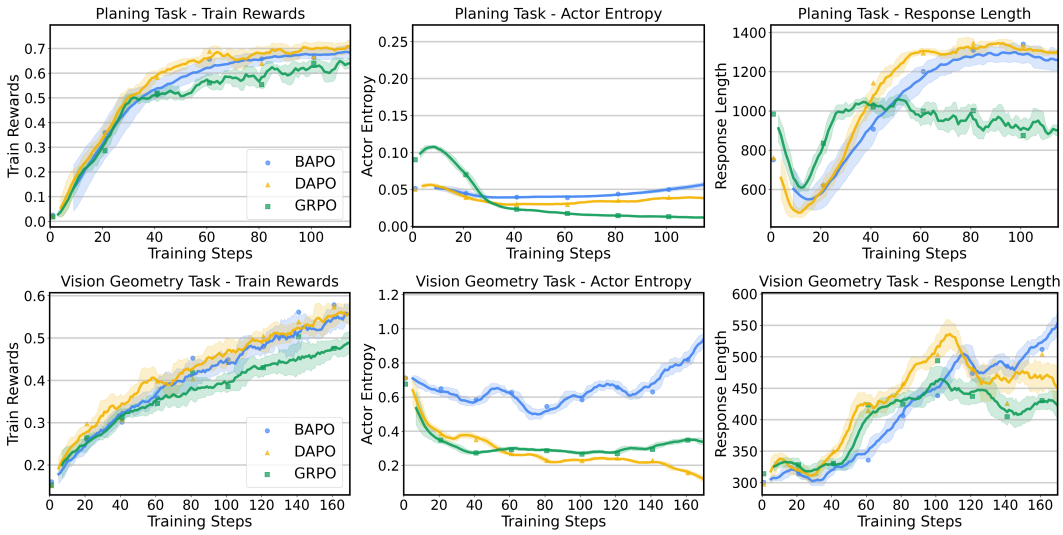

Figure 10: **Training Dynamics** during BAPO, GRPO, and DAPO post-training, including training rewards, training entropy, and response lengths.

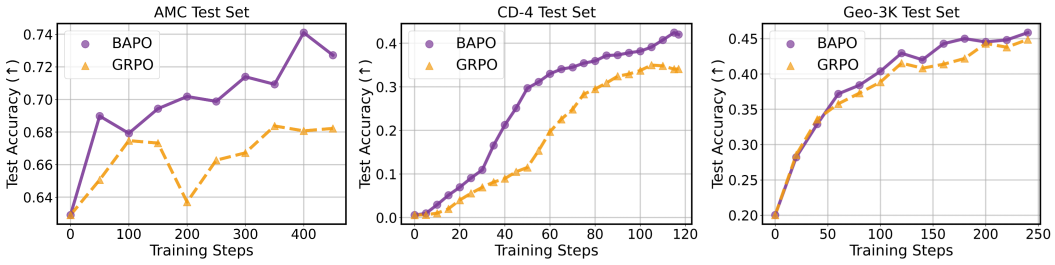

Figure 11: **Test Curves of Group Accuracy Changes** for mathematics, planning, and geometry tasks among AMC, CD-4 test set, and Geo-3K test set, respectively.

A.5 Computation Analysis

From Table 2, we observe that BAPO's computational overhead correlates with the number of samples requiring re-evaluation and the actual training batch size. For the Planning task, BAPO (w/o $\mathcal{X}_2$) achieves the fastest training time by eliminating bad case re-evaluation, but this comes at the cost of reduced performance. For the Mathematics task, the high number of bad cases (as shown by the 0/8 accuracy samples in Figure 7) means that under our re-evaluation frequency setting of $m = 5$, inference time exceeds that of GRPO. However, this additional time investment proves valuable, yielding better bad-case handling rates and overall test performance, as shown in Figure 4 and Table **??**. We plan to explore lower re-evaluation frequencies to assess the performance trade-offs.

BAPO ($c_2 = 0.375, c_3 = 0.5$) runs significantly faster than BAPO ($c_2 = 0, c_3 = 0.25$) due to the larger historical data volume in the latter configuration. This causes BAPO ($c_2 = 0, c_3 = 0.25$) to maintain a larger effective batch size than BAPO ($c_2 = 0.375, c_3 = 0.5$). Training logs also confirm this observation: BAPO ($c_2 = 0, c_3 = 0.25$) consistently utilizes 100% of the configured batch size (equivalent to on-policy methods' batch size), while BAPO ($c_2 = 0.375, c_3 = 0.5$) operates at approximately 70% capacity.

Table 2: **Computational Overhead Analysis.** "**Batch size**" (a, b) represents the sample batch size a and train mini batch size b. "**Time**" is measured in total training time (d=days, h=hours, m=minutes) on 8 A100 GPUs.

Tasks	Methods	Batch Size	Num Epoch	Time
Mathematics	GRPO	(256, 64)	3	**1d 16h 58m**
	DAPO	(256, 64)	3	2d 15h 30m
	BAPO	(256, 64)	3	1d 22h 37m
Planning	GRPO	(256, 64)	3	3h 47m
	DAPO	(256, 64)	3	6h 35m
	BAPO	(256, 64)	3	3h 23m
	BAPO (w/o $\mathcal{X}_2$)	(256, 64)	3	**2h 38m**
	BAPO (w/o $\mathcal{X}_3$)	(256, 64)	3	3h 4m
	BAPO ($c_2 = 0, c_3 = 0.25$)	(256, 64)	3	3h 54m
	BAPO ($c_2 = 0.375, c_3 = 0.5$)	(256, 64)	3	3h 4m
Visual Geometry	GRPO	(256, 64)	30	7h 55m
	DAPO	(256, 64)	30	12h 19m
	BAPO	(256, 64)	30	5h 50m
	BAPO (w/o $\mathcal{X}_2$)	(256, 64)	30	**3h 42m**
	BAPO (w/o $\mathcal{X}_3$)	(256, 64)	30	4h 31m

A.6 Visualization

We present additional visualization details, including the sample accuracy tracking for the Countdown and Geometry3K datasets, as shown in Figure 13. Meanwhile, we visualize the source of

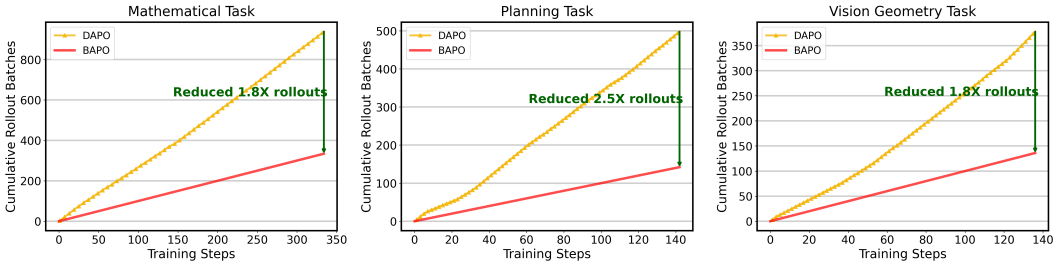

Figure 12: **Cumulative Rollout Batches Comparison between BAPO and DAPO.** The maximum rollout time for DAPO is set to 4.

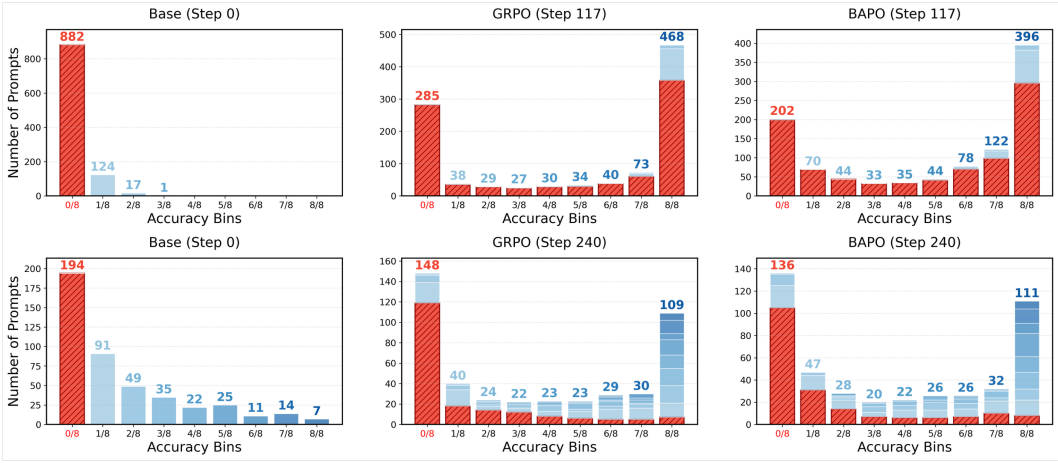

Figure 13: Tracking changes in the **Number of Different Accuracy Bins** on the Countdown (upper) and Geometry3K training sets (lower) for the baseline model, GRPO, and our BAPO method. Special attention is paid to the change in the number of bad samples (red bars) that the base model fails to handle.

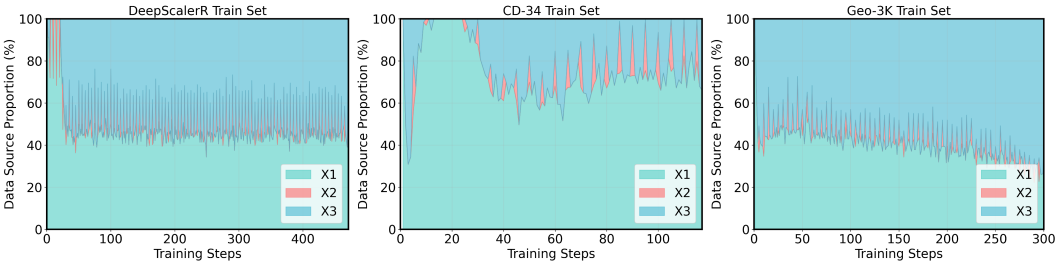

Figure 14: **Batch Distribution Visualization of $\mathcal{X}_1$, $\mathcal{X}_2$, $\mathcal{X}_3$** for Mathematics, Planning, and Visual Geometry Tasks (left to right) during BAPO's training.

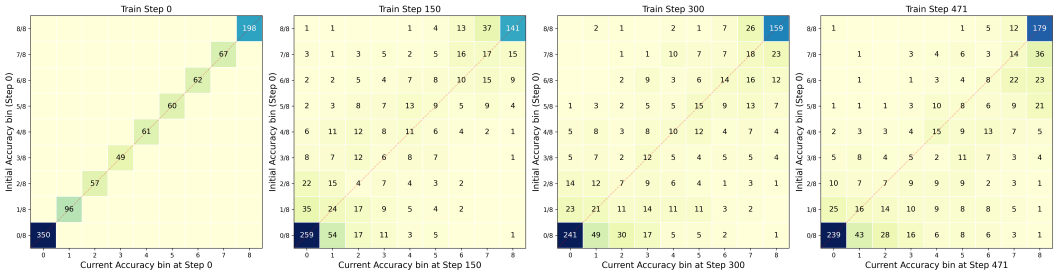

Figure 15: **Accuracy Migration Matrix Analysis.** We track a fixed subset of 1,000 randomly selected prompts from the training set and visualize their movement between accuracy bins (0/8 to 8/8) at Steps 0, 150, 300, and 471 (the last step). The y-axis represents the initial accuracy bin at Step 0, while the x-axis represents the current accuracy bin. **The scarcity of samples in the lower triangle demonstrates that performance degradation is rare.**

samples in each training batch and their respective proportions during the training process, as illustrated in Figure 11. It can be observed that approximately 40-60% of the actual training samples for BAPO come from online samples $\mathcal{X}_1$, while the remaining samples are derived from $\mathcal{X}_2$ or $\mathcal{X}_3$.

Stability of Historical High-Quality Samples. A potential concern regarding the reuse of historical high-quality samples ($\mathcal{X}_3$ in Eq. 5) is the assumption of policy consistency—specifically, whether samples that were high-quality under a past policy remain valid for the current policy. To address this, we visualize the evolution of sample difficulty in Figure 15 by tracking the accuracy migration of a training subset.

The heatmaps in Figure 15 reveal a distinct pattern: the mass is concentrated along the diagonal (performance maintenance) and the upper triangle (performance improvement). Crucially, the proportion of samples exhibiting significant performance degradation (migrating to the lower triangle) is negligible. For example, samples that initially achieved $8/8$ accuracy predominantly remain in the high-accuracy bins throughout the training process, with minimal regression to lower bins. This empirical evidence demonstrates that high-quality reasoning paths learned by RL are robust and resistant to forgetting. Consequently, historical high-quality samples stored in the buffer likely remain high-quality under the current policy, validating the consistency of the $\mathcal{X}_3$ data source.

A.7 Hyperparameter Setting

Hyperparmeters The major hyperparameter choices are shown in Table 3.

Table 3: **Hyperparameter Configuration** for BAPO Framework on Mathematics Task. For planning and visual geometry tasks, some parameters differ slightly; specific configuration scripts are provided in our code repository.

Parameter	Value	Parameter	Value	Parameter	Value
Rollout Configuration					
Top-p	1	Top-k	-1	Temperature	1
Group size (G)	8	Max prompt length	2048	Max response length	8192
Rollout workers	8	Sample batch size	256	Seed	42
Training Configuration					
Learning rate	1e-6	Train mini batch size	64	GAE lambda	1.0
Training epochs	3	KL coefficient (β)	0.001	Entropy coefficient	0.001
Off-policy Configuration					
c_1 threshold	$1/8$	c_2 range	$[1/8, 4/8]$	c_3 range	$[2/8, 5/8]$
Buffer size ($\|B\|$)	256	Rollout delay (v)	5	Re-evaluation freq (m)	5
Gaussian std (σ)	0.2	Gaussian mean (μ)	0.5	Max re-evaluate prompts	128

Reward Function To evaluate the impact of our method, we adopt a simple reward function as below. All training experiments employ the same reward function.

$$r(x, y) = \begin{cases} 1, & \text{if } y \text{ is correct} \\ 0, & \text{otherwise} \end{cases}$$

Datasets and Benchmarks To evaluate the models above, we use three training datasets and eight benchmarks categorized into mathematical, planning and vision geometry reasoning benchmarks as described in Table 4.

Table 4: **Datasets and Benchmarks** used in this study.

Dataset	#Train	#Test	Task Type	Domain	License	Source
Training Datasets						
DEEPSCALER-1.5B-PREVIEW	40,000	–	Math reasoning	Mathematics	Apache 2.0	Link
COUNTDOWN-TASKS-3TO4	49,000	–	Logic reasoning	Planning	Apache 2.0	Link
GEOMETRY3K	2,100	–	Visual reasoning	Visual Geometry	Apache 2.0	Link
Test Benchmarks						
AIME24	–	30	Math competition	Mathematics	MIT	Link
AMC	–	83	Math competition	Mathematics	Apache 2.0	Link
MATH500	–	500	Math reasoning	Mathematics	-	Link
MINERVA	–	272	Math reasoning	Mathematics	Apache 2.0	Link
OLYMPIAD	–	674	Math competition	Mathematics	Apache 2.0	Link
COUNTDOWN-TASKS-3TO4	–	200*	Logic reasoning	Planning	Apache 2.0	Link
COUNTDOWN-TASKS-4	–	200*	Logic reasoning	Planning	Apache 2.0	Link
GEOMETRY3K	–	901	Visual reasoning	Visual Geometry	Apache 2.0	Link

*We only use a random subset of this benchmark for faster ablation studies.

A.8 ALGORITHM

Algorithm 1 presents the proposed BAPO, which can be seamlessly integrated with any GRPO-like RLVR algorithm.

Algorithm 1 Batch Adaptation Policy Optimization (BAPO)

Require: Policy π_{θ_0}, buffer $\mathcal{B} = \emptyset$, thresholds c_1, c_2, c_3, delay steps v, re-evaluate frequency m

for $t = 1$ to T **do**
  // Off-policy Rollout Phase
  if $t \bmod v = 0$ **then**
    Synchronize rollout policy's parameter with trainer: $\alpha = \pi_{\theta_t}$
  end if
  Using rollout policy α to generate G responses $\{y_j\}_{j=1}^{G}$ for each question x
  Compute log probabilities $\alpha(y|x)$ and rewards r for constructing the online batch $\mathcal{X}_{\text{on}}$
  Store samples into buffer $\mathcal{B}_{\text{bad}} \leftarrow \{(x, y, \alpha(y|x), r) \in \mathcal{X}_{\text{on}} : \mu_{\alpha,r}(x) \leq c_1\}$
  Store samples into buffer $\mathcal{B}_{\text{high}} \leftarrow \{(x, y, \alpha(y|x), r) \in \mathcal{X}_{\text{on}} : c_2 \leq \mu_{\alpha,r}(x) \leq c_3\}$
  // Off-policy Training Phase
  $\mathcal{X}_1 \leftarrow$ online filter on $\mathcal{X}_{\text{on}}$ with $\mu_{\alpha,r}(x) \in \{\frac{1}{G}, \ldots, \frac{G-1}{G}\}$ (Filtered Fresh Samples)
  $\mathcal{X}_2 \leftarrow \emptyset$
  if $t \bmod m = 0$ **then**
    Re-evaluate $\mathcal{B}_{\text{bad}}$ with π_{θ_t} to get $\mathcal{X}_2$ using Equation 6 (Re-evaluated Difficult Samples)
  end if
  $\mathcal{X}_3 \leftarrow$ Sample from $\{(x, y) \in \mathcal{B}_{\text{high}} : \mu_{\alpha_{\mathcal{B}},r}(x) \in [c_2, c_3]\}$ (Historical High-quality Samples)
  Final batch $\leftarrow \mathcal{X}_1 \cup \mathcal{X}_2 \cup \mathcal{X}_3$
  Compute advantages and update critic/actor with final_batch
  Add $\mathcal{D}_t$ to buffer $\mathcal{B}$
end for

A.9 Generalization Analysis

To demonstrate the algorithmic generalizability of our framework, we extended the Batch Adaptation paradigm to Proximal Policy Optimization (PPO), denoted as **BA-PPO**. In this experiment, both the Actor and Critic networks were initialized with the **Qwen3-4B** backbone and trained on the **DeepScaleR** dataset with a maximum response length of 4K tokens. We maintained consistency with the foundational BAPO configuration by applying standard zero-advantage filtering for $\mathcal{X}_1$ (removing only all-correct and all-wrong groups), utilizing the initial BAPO values for thresholds c_1, c_2, c_3, and setting the buffer size to 64.

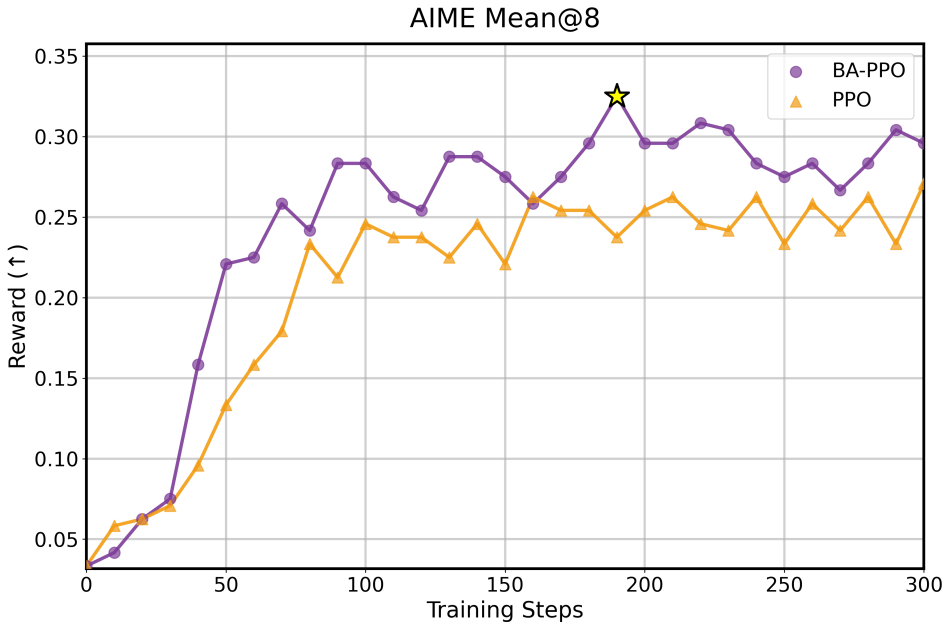

Figure 16: **Generalization to Actor-Critic Algorithms (BA-PPO).** Performance comparison between standard PPO (orange triangles) and BA-PPO (purple circles) on the AIME 2024 benchmark using Qwen3-4B. The star ($\star$) marks the peak performance of BA-PPO (0.325).

As illustrated in Figure 16, BA-PPO achieved a remarkable performance gain of **+5.5** on the **AIME 2024 benchmark** compared to the standard PPO baseline. This result further confirms that the core principle of dynamic batch construction is effective not only for GRPO but also functions as a robust, algorithm-agnostic enhancement for actor-critic methods.