LEARNING WITH A SINGLE ROLLOUT VIA PASS@K POLICY OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Group Relative Policy Optimization (GRPO) typically requires multiple rollouts per prompt to estimate relative credit, incurring substantial sampling cost and exacerbating training asynchrony. This limitation is particularly severe for scientific reasoning agents, where each trajectory may involve long-horizon reasoning, multiple turns, and parallel sub-runs. We study *single-rollout* reinforcement learning, where only one rollout is collected per prompt, and find that naive training can stagnate or become severely unstable. We introduce Single-Rollout Pass@k Policy Optimization (SR-PPO), which combines Monte Carlo token-level advantage estimation with Pass@k gradient reweighting. An online critic without large-scale pretraining is used both as a prefix-dependent baseline for credit assignment and to estimate the Pass@k gradient weight. Using only one rollout per prompt, SR-PPO achieves stable learning and performance comparable to GRPO across reasoning and logic benchmarks, even under off-policy drift that degrades the performance of standard Pass@1 PPO. A key factor in this stability is Pass@k reweighting, which emphasizes harder prompts while downweighting high-success prompts, for which imperfect credit estimates tend to produce noisier policy updates. Through analyses of critic quality, gradient noise, and update behavior, we investigate how Pass@k reweighting stabilizes single-rollout learning and why residual critic errors may limit SR-PPO's ability to outperform GRPO.

1 INTRODUCTION

Language models (LMs) have made rapid progress in complex reasoning, particularly when post-trained with reinforcement learning (RL) from verifiable rewards (Cobbe et al., 2021; Hendrycks et al., 2021; Lewkowycz et al., 2022; Shao et al., 2024; Guo et al., 2025). However, policy gradient estimates based on binary outcome rewards can exhibit high variance, especially under high-entropy sampling used to encourage exploration. To stabilize optimization, Group Relative Policy Optimization (GRPO) samples multiple responses to each prompt and constructs advantages by normalizing rewards within the group (Shao et al., 2024). This simple variance-reduction mechanism has made GRPO a widely adopted recipe for RL post-training.

Yet its reliance on multiple rollouts introduces substantial sampling cost. This limitation becomes particularly important for modern agentic systems, where a single trajectory may involve long-horizon reasoning, tool calls (Wu et al., 2026), memory retrieval (Jia et al., 2025), sub-agents (Wang et al., 2025), or expensive environment interactions (Yang et al., 2026). Moreover, independently sampled trajectories can quickly diverge into qualitatively different prefixes, making cross-rollout comparison increasingly indirect (Hou et al., 2026).

These limitations motivate the *single-rollout* setting, in which only one solution is generated for each prompt. Single-rollout RL removes the requirement of repeatedly sampling the same problem, but also removes the group-relative baseline that makes GRPO stable. Consequently, estimating token-level credit in a single rollout that yields low-variance advantages becomes a central challenge. A natural solution is to introduce a critic that predicts the return starting from each prefix, which is the probability of eventual success from intermediate prefixes in the outcome-reward setting. Then, temporal changes in the critic value can then be used to construct token-level advantages.

Standard PPO uses such an auxiliary critic for credit assignment (Schulman et al., 2015; 2017). In our experiments, however, the critic produces unreliable token-level credit estimates, which con-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

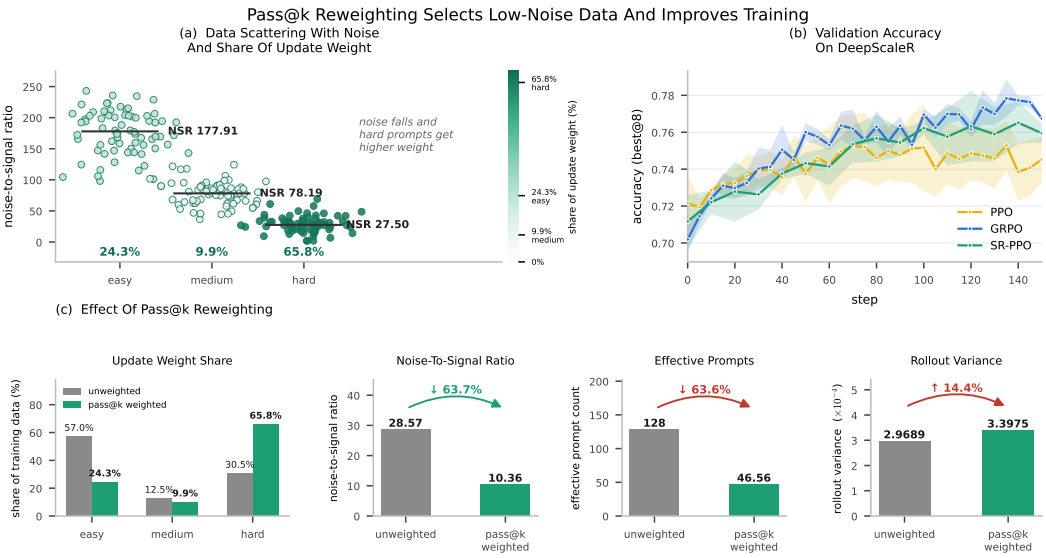


Figure 1: **Method overview and performance for SR-PPO. (a)** Hard prompts are defined to have at most two correct rollouts among eight independent samples from the given checkpoint. Noise-to-signal ratio measures the noise level of the gradient. The ratio reduces as the prompts get harder. Our SR-PPO gives least noisy data the highest share in the gradient update and thus stabilizes learning. **(b)** Validation Best@8 success rates of our SR-PPO, GRPO and PPO versus training steps under a fair computation comparison. **(c)** After adding Pass@k reweighting, hard prompts get more shares in the gradient and also noise-to-signal ratio reduces. However, the variance of gradient even increases with less effective prompts per batch, implying that noise-to-signal ratio reduction can be the true reason for performance improvement.

tribute to the poor performance and high variability across seeds of PPO observed in Figure 1(b). Extensive critic pretraining would require additional rollouts and does not ensure that value estimates remain accurate as the policy's rollout distribution evolves. We therefore train the critic online using the same single-rollout data collected for policy optimization.

This choice introduces a central tension: online learning allows the critic to track an evolving policy, but its prefix-value estimates may be unreliable, particularly early in training. We address this challenge with Single-Rollout Pass@k Policy Optimization (SR-PPO), which combines Monte Carlo advantage estimation with Pass@k reweighting to improve robustness to imperfect credit estimates. SR-PPO uses the critic only as a variance-reducing baseline, while the final outcome reward is still used as the unbiased return estimator. Meanwhile, we optimize the Pass@k objective, where Pass@k denotes the probability that at least one of k independent continuations succeeds. With a learnt critic, we derive a novel single-rollout estimator of the Pass@k policy gradient with a prompt reweighting term. As shown in Figure 1(c), this estimator substantially reduces gradient noise-to-signal ratio, leading to steady improvement comparable to multi-rollout GRPO in Figure 1(b).

A key property of the Pass@k gradient is its prompt-dependent weighting: it suppresses updates from high-success prompts while placing greater emphasis on harder prompts. Empirically, harder prompts yield less noisy gradient estimates as shown in Figure 1(a), so this reweighting increases the contribution of more reliable training data and thus stabilizes the training. We hypothesize that easy prompts produce noisier gradients because failed rollouts on nearly solved problems are often mostly correct except for a few decisive mistakes, making accurate error localization especially important. If the critic misjudges these failure points, it can assign positive credit to unrecoverable mistakes. Meanwhile, downweighting easy prompts is unlikely to impair learning, as their already high success rates leave limited room for further improvement. Pass@k therefore acts not only as an alternative policy objective, but also as a mechanism for controlling critic-induced gradient noise in single-rollout learning. More analysis is provided in Section 5.5.

We investigate this mechanism through analyses of critic quality and gradient behavior. The critic broadly tracks prompt difficulty, assigning higher values to easier prompts in agreement with group-mean rewards, shown in Table 6, yet performs near chance at localizing mistakes within individual

rollouts, shown in Table 5. This helps explain both the source of gradient noise and why improved training stability alone may not enable SR-PPO to outperform GRPO. Our gradient analysis further reveals that easy-prompt updates can partially cancel contributions from hard prompts in Figure 4, even though these updates remain largely concentrated in the predefined safe region of the pretrained weights (Zhu et al., 2025). These findings suggest that update location alone does not explain training collapse and highlight gradient interference as a potential contributing factor.

Under off-policy training, SR-PPO improves over Pass@1 PPO on competition-level mathematical reasoning. A smaller-batch variant also sustains stable learning through step 600. Across AIME 24-26 (Zhang & Math-AI, 2024; 2025; 2026), HMMT 26 (Dekoninck et al., 2026b), Knights & Knaves (Xie et al., 2025), GPQA-D (Rein et al., 2024), and Countdown4 (Park et al., 2026), SR-PPO remains broadly competitive with GRPO, with performance differences small relative to seed variations. Although imperfect critic estimates may still limit attainable performance, these results support SR-PPO as a practical training recipe for stable improvement using only one rollout per prompt.

2 BACKGROUND

Given a prompt x, a policy π_θ generates a token sequence $y_{1:T}$ of length T. We define the prefix state as $s_t = (x, y_{1:t})$, with $s_0 \sim d$ containing only the prompt, d being the data distribution and $y_t \sim \pi_\theta(\cdot \mid s_{t-1})$. A verifier returns a binary outcome reward $Y : s_T \mapsto \{0, 1\}$, inducing rewards $R_t = 0$ for $t < T$ and $R_T = Y$.

Value Function Estimation Value function is defined to be the expected return starting from the current prefix. In this undiscounted setting, the value equals the probability of eventual success:

$$q_\pi(s_t) := \mathbb{E}_\pi\left[\sum_{u=t+1}^{T} R_u \,\middle|\, s_t\right] = \mathbb{E}_\pi[\mathbb{1}[Y = 1] \mid s_t] = \mathbb{P}_\pi(Y = 1 \mid s_t). \tag{1}$$

For an integer $k \geq 1$, Pass@k value is the probability that at least one of k independent continuations succeeds (Walder & Karkhanis, 2026), defined as

$$q_{\pi,k}(s_t) := \mathbb{P}_\pi\left(\max_{1\leq i\leq k} Y_i = 1 \,\middle|\, s_t\right) = 1 - \big(1 - q_\pi(s_t)\big)^k, \tag{2}$$

where the outcomes $Y_1, \ldots, Y_k$ are conditionally independent given s_t and sampled under π. Here, k specifies the optimization target, not the training rollout count.

Since prefix transitions are deterministic, the token-level advantage is defined to be

$$A_\pi(s_{t-1}, y_t) := q_\pi(s_{t-1}, y_t) - q_\pi(s_{t-1}) = q_\pi(s_t) - q_\pi(s_{t-1}), \tag{3}$$

which measures a token's contribution to eventual success relative to the expected continuation.

The final binary outcome Y is a Monte Carlo estimator for $q_\pi(s_t)$. Also, we can introduce an extra critic to estimate the value function, denoted as $\hat{q}_\phi$. This gives the Monte Carlo advantage estimator:

$$\widehat{A}_t^{\mathrm{MC}} = Y - \hat{q}_\phi(s_{t-1}), \tag{4}$$

where the estimated value function serves only as a pre-token baseline.

With a learned critic, an alternative is to construct the advantage directly from temporal differences:

$$\widehat{A}_t^{\mathrm{TD}} = \hat{q}_\phi(s_t) - \hat{q}_\phi(s_{t-1}). \tag{5}$$

However, this estimator can inherit substantial bias from value-function approximation errors, while reducing variances from the randomness of the outcome reward.

PPO utilizes generalized advantage estimation (Schulman et al., 2015), which interpolates between temporal-difference estimates and Monte Carlo returns through the parameter λ. In the undiscounted setting, setting $\lambda = 1$ recovers the Monte Carlo estimator above. Thus, the Monte-Carlo advantage estimator can also be denoted as $\widehat{A}_t^{\mathrm{GAE}(1)}$, which is the default PPO implementation.

Group Relative Policy Optimization. GRPO replaces the critic with reward normalization across G responses to the same prompt (Shao et al., 2024). Let $\overline{Y}$ and σ_Y denote the mean and standard deviation of their outcomes. Every token in response i receives the same advantage:

$$\widehat{A}_i^{\mathrm{GRPO}} = \frac{Y_i - \overline{Y}}{\sigma_Y + \varepsilon_{\mathrm{std}}}, \quad 1 \le i \le G, \tag{6}$$

where $\varepsilon_{\mathrm{std}} > 0$ ensures numerical stability. For $G = 1$, the advantage is zero, requiring a different estimator for single-rollout learning.

Using trajectories sampled from the behavior policy π_{old}, we maximize a clipped surrogate that limits the incentive for large changes in token probabilities (Schulman et al., 2017; Shao et al., 2024). Omitting the KL regularization to the base model, the GRPO objective (Liu et al., 2025) is

$$\mathcal{J}(\theta) = \mathbb{E}_{\substack{s_0 \sim d, \\ y^{(i)} \sim \pi_{\mathrm{old}}(\cdot|s_0)}} \left[\frac{1}{G} \sum_{i=1}^{G} \sum_{t=1}^{T_i} \min\left\{ \rho_{i,t}(\theta) A_i^{\mathrm{GRPO}}, \mathrm{clip}(\rho_{i,t}(\theta), 1-\epsilon, 1+\epsilon)\, A_i^{\mathrm{GRPO}} \right\} \right], \tag{7}$$

where G is the group size, T_i is the length of response i, and the importance sampling ratio to correct the distribution shift is defined as $\rho_{i,t}(\theta) = \dfrac{\pi_\theta(y_t^{(i)} \mid s_{t-1}^{(i)})}{\pi_{\mathrm{old}}(y_t^{(i)} \mid s_{t-1}^{(i)})}$.

Pass@k Policy Optimization. With the defined value function, standard expected-return policy optimization maximizes Pass@1, with the objective written as $J_\pi(\theta) = \sum_{s_0} d(s_0) q_\pi(s_0)$.

Pass@k optimization instead maximizes the probability that at least one of k independently sampled solutions succeeds, with the following objective:

$$J_{\pi,k}(\theta) := \sum_{s_0} d(s_0) q_{\pi,k}(s_0) = \sum_{s_0} d(s_0) \left[1 - \big(1 - q_\pi(s_0)\big)^k \right]. \tag{8}$$

Also, Dekoninck et al. (2026a) establish the transformation between the Pass@1 and Pass@k value functions, which yields the second equality in our objective. We state the corresponding token-level policy gradient theorem in Section 4 and use it to construct a PPO-style surrogate objective.

3 Related Work

Single-Rollout RL A recent paper develops Single-Rollout Asynchronous Optimization (SAO) which studies the limitations of group-wise sampling in asynchronous agentic RL (Hou et al., 2026) and explores the single-rollout setup. SAO includes more frequent critic updates, frozen attention modules, observation-skipping GAE, and scaled value pretraining to stabilize training. SAO shares our motivation for avoiding repeated sampling. However, our emphasis is on learning with an online critic without large-scale critic pretraining so that it can adapt to frontier reasoning tasks and study the effect of Pass@k gradient weighting in stablizing training.

Dense Credit Assignment. Several methods exploit structure across sampled trajectories to obtain fine-grained credit. TEMPO converts groups of completions into prefix trees, estimates nonparametric values from descendant outcomes, and augments GRPO with branch-gated temporal-difference corrections (Tran et al., 2025). TreePO combines dynamic tree sampling, fixed-length segment decoding, and tree-based segment-level advantages to share computation across prefixes while preserving exploration (Li et al., 2025). For long-horizon agents, GiGPO combines trajectory-level advantages with action-level comparisons at repeated anchor states (Feng et al., 2025), while HGPO aggregates advantages across hierarchical groups to address historical-context inconsistency (He et al., 2026). VIMPO instead offers a critic-free but value-based approach by deriving policy-implied values from the optimality conditions of KL-regularized RL (Kang et al., 2026). These methods improve credit assignment within a multi-rollout setting, whereas our target setting does not provide the branching or grouping structure they use.

Another growing line of work studies process-level reward modeling for reasoning (Cobbe et al., 2021; Lightman et al., 2024; Wang et al., 2024; Cui et al., 2025). AgentPRM extends progress-based evaluation to agentic settings, where intermediate actions are assessed by their contribution

toward the final goal rather than by isolated correctness (Xi et al., 2026). Likelihood-based rewards provide another form of dense supervision by rewarding the probability assigned to a reference answer or continuation (Kwiatkowski et al., 2026). These approaches primarily focus on learning or constructing informative intermediate reward signals. In contrast, our focus is on estimating token-level policy advantages, which determines the direction and magnitude of policy updates. Rather than improving the token-level credit assignment itself, we further study how to stabilize policy optimization when credit estimates are imperfect, particularly in the single-rollout regime.

Pass@k Policy Optimization. Several studies directly optimize multi-sample success objectives such as Pass@k sucess rate to encourage exploration (Chen et al., 2025; Takashiro et al., 2026). Pass-at-k Policy Optimization (PKPO) derives unbiased estimators for Pass@k policy gradients using more than k samples per prompt (Walder & Karkhanis, 2026). Beyond Accuracy derives a direct policy-gradient reweighting that emphasizes low-success prompts but the estimator still requires multiple rollouts (Mahdavi et al., 2025). Advantage Shaping as Surrogate Reward Maximization connects direct Pass@k gradients with GRPO-style advantage shaping through their underlying population objectives without introducing any new estimators, interpreting hard-example reweighting as reward-level regularization (Thrampoulidis et al., 2026). Our contribution is a new Pass@k gradient estimator formulation in single-rollout learning with an online prefix critic. We further study how Pass@k reweighting interacts with token-level credit assignment to improve robustness to critic error, without requiring additional per-prompt rollouts or large-scale critic pretraining.

4 METHOD

This section presents our single-rollout training framework, which combines Pass@k policy optimization with Monte Carlo advantage estimation. For completeness, we restate the token-level policy gradient theorem with its Pass@k extension, specializing established derivations to undiscounted setting with binary terminal rewards. We then introduce a critic-based estimator for the Pass@k policy gradient, formulate the corresponding PPO-style clipped surrogate (Schulman et al., 2017), and describe the critic design.

4.1 TOKEN-LEVEL PASS@K POLICY GRADIENT

We follow the trajectory-level likelihood-ratio derivation of REINFORCE (Willams, 1992). This formulation operates directly on complete rollouts and their final outcome rewards, with prefix-visitation probabilities implicit in the trajectory expectation. Subtracting a prefix-dependent baseline from the outcome reward yields an unbiased gradient estimator with Monte-Carlo advantage introudced in Equation 4. The Pass@k policy gradient is obtained based on the established gradient reweighting (Takashiro et al., 2026), by differentiating the probability transformation in Equation 2. The following lemma states a token-level Pass@k policy gradient and provides the foundation for single-rollout policy gradient estimator.

Lemma 4.1 (Token-Level Pass@k Policy Gradient)**.** *Let $\pi = \pi_\theta$ be a differentiable autoregressive policy with bounded trajectory lengths T. Given the prompt distribution d and the binary reward $Y(s_0, y_{1:T}) \in \{0, 1\}$, for an integer $k \geq 1$ and any bounded suffix-independent baseline b, the Pass@k policy gradient satisfies*

$$\nabla_\theta J_{\pi,k}(\theta) = \mathbb{E}\left[w_{\pi,k}(s_0) \sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(y_t \mid s_{t-1}) \big(Y - b(s_{t-1})\big) \right], \tag{9}$$

where the expectation is over $s_0 \sim d$ and $y_{1:T} \sim \pi_\theta(\cdot \mid s_0)$, and

$$w_{\pi,k}(s_0) := k\big(1 - q_\pi(s_0)\big)^{k-1}.$$

For $k > 1$, the weight $w_{\pi,k}(s_0)$ in the policy gradient decreases with prompt success probability, assigning greater relative weight to harder prompts. The proof is written in Appendix A.

4.2 PASS@K GRADIENT ESTIMATOR WITH MONTE-CARLO ADVANTAGE

Our algorithm relies on an extra trained critic, denoted as $\hat{q}_\phi$. With the critic, the Pass@k weights can be directly estimated as $\hat{w}_{\pi,k}(s_0) = k\big(1 - \hat{q}_\phi(s_0)\big)^{k-1}$, without needing multiple rollouts to

approximate the Pass@k success rate. Also, we instantiate the baseline with the learned prefix critic, $b(s_{t-1}) = \hat{q}_\phi(s_{t-1})$, obtaining the Monte Carlo advantage estimator shown in Equation 4. Here, the value function serves only as a baseline and the return is supplied by the observed outcome.

Given a batch $\mathcal{B}$ of prompts, SR-PPO first computes the Monte Carlo advantages and plug-in Pass@k weights for each prompt i, denoted as $\widehat{w}_{i,k} = k\big(1 - \hat{q}_\phi(s_0^{(i)})\big)^{k-1}$ and $\widehat{A}_{i,t}^{\mathrm{MC}} = Y^{(i)} - \hat{q}_\phi(s_{t-1}^{(i)})$. Then, the gradient estimator for Pass@k equals

$$\frac{1}{|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|}\widehat{w}_{i,k}\sum_{t=1}^{T^{(i)}}\nabla\log\pi_\theta(y_t^{(i)}|s_{t-1}^{(i)})\big(Y^{(i)} - \hat{q}_\phi(s_{t-1}^{(i)})\big). \tag{10}$$

Our method adopts PPO-style gradient clipping (Schulman et al., 2017; Shao et al., 2024) when collecting rollouts under π_{old} in the off-policy setting. The surrogate policy optimization objective for gradient clipping equals

$$\mathcal{L}_{\mathrm{actor}}(\theta) = -\frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}\widehat{w}_{i,k}\sum_{t=1}^{T_i}\min\Big\{\rho_{i,t}(\theta)\widehat{A}_{i,t}^{\mathrm{MC}}, \mathrm{clip}(\rho_{i,t}(\theta), 1-\epsilon, 1+\epsilon)\,\widehat{A}_{i,t}^{\mathrm{MC}}\Big\}, \tag{11}$$

with the importance sampling ratio $\rho_{i,t}(\theta) = \dfrac{\pi_\theta(y_t^{(i)}|s_{t-1}^{(i)})}{\pi_{\mathrm{old}}(y_t^{(i)}|s_{t-1}^{(i)})}$ and the clipping parameter $\epsilon > 0$.

4.3 Critic Design

When training our separate critic network, denoted as $\hat{q}_\phi$, each nonterminal prefix is supervised by the observed binary outcome of its completed trajectory. Given a rollout batch $\mathcal{B}$, the critic is trained with binary cross-entropy augmented by an auxiliary mean-squared-error term:

$$\begin{aligned}\mathcal{L}_{\mathrm{critic}}(\phi;\mathcal{B}) = \frac{1}{\sum_{i\in\mathcal{B}}T^{(i)}}\sum_{i\in\mathcal{B}}\sum_{t=1}^{T^{(i)}}\Big[&-Y^{(i)}\log\hat{q}_\phi(s_{t-1}^{(i)}) - (1-Y^{(i)})\log\big(1-\hat{q}_\phi(s_{t-1}^{(i)})\big)\\ &+\lambda_{\mathrm{cal}}\big(\hat{q}_\phi(s_{t-1}^{(i)}) - Y^{(i)}\big)^2\Big],\end{aligned} \tag{12}$$

where $\lambda_{\mathrm{cal}} \geq 0$ controls the calibration level so that the logit output represents the true probability value. The critic is trained online using the same rollout data collected for policy optimization but after the actor update to get an effective baseline.

SR-PPO differs from our PPO baseline in both Pass@k weighting and critic design. Whereas the PPO baseline trains an unconstrained value head using mean-squared error alone, SR-PPO uses a sigmoid output and the combined binary cross-entropy and squared-error loss. The sigmoid parameterization ensures that $\hat{q}_\phi(s) \in (0,1)$, allowing critic predictions to be used directly as success-probability estimates in Pass@k reweighting.

5 Experiment Results

We evaluate SR-PPO against PPO and GRPO, with particular emphasis on whether Pass@k reweighting stabilizes learning under misleading credit assignment induced by an imperfect critic. All evaluation metrics in this section are introduced in Appendix C.

5.1 Experimental Setup

We conduct mathematical reasoning experiments using Qwen3-1.7B (Yang et al., 2025) in thinking mode as the base model, with a maximum response length of 9,182 tokens. Training is performed on the DeepScaleR mathematical reasoning dataset (Luo et al., 2025). Our goal is to identify a stable single-rollout credit-assignment method that improves the reasoning capabilities of a small open-source base model, rather than to establish the advantages of single-rollout training in agentic tasks, as investigated by Hou et al. (2026).

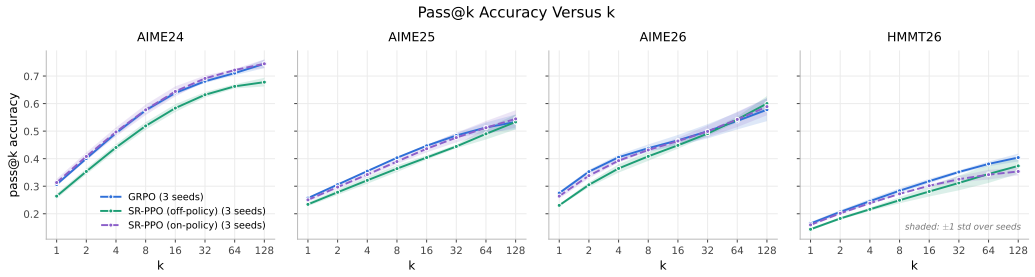


Figure 2: Pass@k accuracy on four competition-math benchmarks. GRPO and SR-PPO (off-policy) are evaluated at step 150, SR-PPO (on-policy) at step 600. Each benchmark has 30 prompts (AIME) or 33 (HMMT26), so a single problem is worth roughly 3 accuracy points.

To approximately match per-update computation, GRPO uses a mini-batch size of 64 with 4 rollouts per prompt, whereas SR-PPO and PPO use a mini-batch size of 128 with a single rollout per prompt. For SR-PPO, we halve the effective actor mini-batch size when accounting for the additional critic update cost. Other training hyperparameters are provided in the Appendix B.

5.2 Math Reasoning Perforamnce

Our goal is to identify a stable single-rollout credit-assignment method that improves the reasoning capabilities of a small open-source base model, rather than to establish the advantages of single-rollout training in agentic tasks, as investigated by Hou et al. (2026).

In the on-policy setting, the train data batch size is used to perform one gradient update per step and thus all importance sampling ratio remains one, without triggering the clipping, while the off-policy algorithm performs four gradient updates per step. Thus, the on-policy SR-PPO is evaluated at step 600, which gives the same number of gradient updates.

All three methods show the expected monotone increase in Pass@k with k, rising steeply from k = 1 to k = 16 and flattening thereafter as the estimator exhausts the 128 available rollouts. On-policy SR-PPO and GRPO are close to indistinguishable on four benchmarks, while off-policy SR-PPO is a weaker configuration. The largest gap between off-policy SR-PPO and GRPO is on AIME24, where it trails GRPO by about 4% at pass@1 and 7% at pass@128. Its one favourable result is on AIME26 at large k, where it overtakes other two methods. In conclusion, no method is separated from the others by a margin that clearly exceeds seed and sampling noise.

5.3 Held-Out Reasoning and Logic Test

Metric	Knights & Knaves	GPQA-D	Countdown4
Base Model	100.00	46.46	48.44
GRPO	100.00 ± 0.00	65.49 ± 3.55	$\mathbf{56.90 \pm 2.35}$
On-Policy SR-PPO	100.00 ± 0.00	$\mathbf{67.68 \pm 2.67}$	55.34 ± 2.60
Off-Policy SR-PPO	100.00 ± 0.00	61.28 ± 4.85	55.08 ± 2.71

Table 1: Best@16 success rate on three held-out reasoning benchmarks, reported as mean $\pm$ standard deviation over three seeds. The base model is evaluated once and has no seed variation. Knights & Knaves is saturated at 100% for all models.

To assess whether improvements from mathematical training transfer beyond the training distribution, we evaluate all methods on three held-out benchmarks (Table 1), introduced in Appendix D.1. Knights & Knaves is saturated, where the base model already solves every instance. On the two discriminative benchmarks, every trained method improves substantially over the base model, confirming that the reasoning ability acquired during math training is not confined to the training domain. Among the trained methods the differences are small relative to seed variation. On-policy SR-PPO attains the highest GPQA-D accuracy, while GRPO is nominally best on Countdown4, but the standard deviations overlap substantially. We therefore read these results as showing that on-policy SR-PPO transfers on par with GRPO.

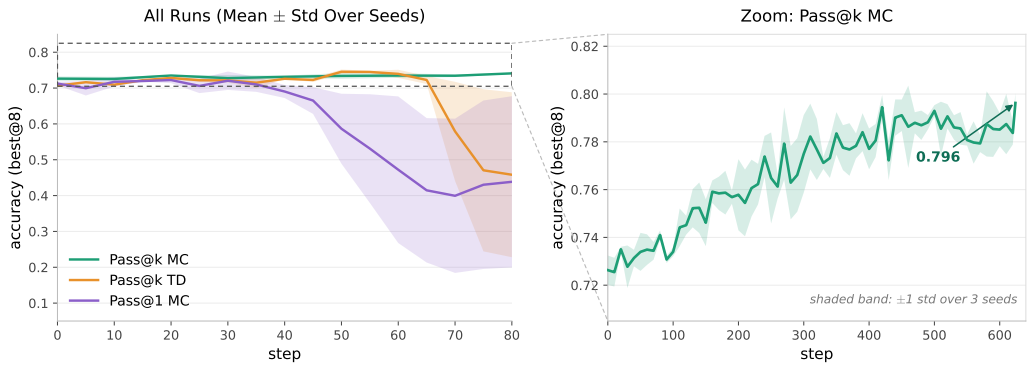


Figure 3: **Ablation study of on-policy SR-PPO. (a)** Both Monte Carlo advantage estimation and Pass@k optimization are important for stable single-rollout training. Using instead a temporal-difference estimator or Pass@1 optimization causes training to collapse around step 80. **(b)** SR-PPO remains stable over two epochs of prolonged training.

5.4 On-Policy SR-PPO Variant

Figure 3(a) shows that both Monte Carlo advantage estimation and Pass@k reweighting are important for stable single-rollout learning, even in the on-policy setting. Replacing it with one-step temporal-difference advantages causes training to collapse, suggesting that critic prediction errors can introduce substantial bias into the policy gradient. The noise-to-signal reduction effect from Pass@k reweighting is thoroughly studied in the next subsection. Figure 1(b) shows that SR-PPO learns steadily in a prolonged training to 600 steps.

5.5 Gradient Noise Analysis

All the critic quality and noise analysis are performed using a checkpoint from on-policy SR-PPO at step 180, while the rollouts are generated from a GRPO policy.

To quantify the stability of stochastic policy-gradient estimates, we measure the noise-to-signal ratio (NSR), defined as the gradient variance relative to the squared norm of the mean gradient. Given R independent rollout replicas with gradients

$$g^{(r)} = \mu + \varepsilon^{(r)},\ 1 \le r \le R, \qquad \mu = \mathbb{E}[g^{(r)}],$$

the population NSR is

$$\mathrm{NSR} = \frac{\mathbb{E}\left[\|\varepsilon\|_2^2\right]}{\|\mu\|_2^2} = \frac{\mathrm{tr}\,\mathrm{Cov}(g)}{\|\mathbb{E}[g]\|_2^2}.$$

A lower NSR indicates that stochastic gradients are more concentrated around their expected update direction, while a higher NSR indicates stronger rollout-induced noise. The finite-sample estimator for NSR is detailed in Appendix C.

Figure 1(c) reports NSR for the gradient aggregated over the full batch, while Figure 1(a) and Table 2 analyze gradients at the prompt level. Pass@k reweighting reduces the batch-level NSR from 28.57 to 10.36, a 63.7% reduction, providing a direct explanation for its improved training stability.

Metric	Easy	Medium	Hard
Number of Prompts	73	16	39
SR-PPO per-prompt NSR mean	177.915	78.188	27.496
SR-PPO per-prompt NSR median	55.16	49.02	15.53
SR-PPO Safe BF16 Changes	99.920%	99.894%	99.893%

Table 2: **Per-prompt gradient noise-to-signal ratio and parameter-update analysis by problem difficulty.** Prompts are grouped according to the number of successful responses among eight rollouts: easy (≥ 6), medium (3–5), and hard (≤ 2). SR-PPO exhibits its lowest NSR on hard prompts. The last row report the fraction of BF16-realized parameter changes that fall within the safe region.

To understand where this reduction comes from, we group prompts according to their empirical difficulty. We classify a prompt as *hard* if at most two rollouts are correct, *medium* if three to five are correct, and *easy* if at least six are correct. For SR-PPO the hard prompts provide the least noisy updates, as shown in Table 2. Consistent with this observation, Pass@k reweighting increases the fraction of total update weight assigned to hard prompts from 30.5% to 65.8%, while reducing the contribution of easy prompts from 57.0% to 24.3%, shown in Figure 1(c). Thus, Pass@k does not merely rescale the gradient; it changes the effective training distribution toward prompts whose SR-PPO gradients have substantially lower NSR ratio.

We further investigate whether noisy gradients destabilize training by modifying parameters associated with the principal structure of the pretrained model. We follow the definition of a *safe region* in Zhu et al. (2025) that excludes parameters with large contributions to the principal low-rank component while retaining low-magnitude parameters. For an parameter update, we measure the fraction of BF16-realized parameter changes that fall within this region and the details of the evaluation metric is introduced in Appendix C.

We find that all difficulty groups place near 100% fraction of their change in the safe region, whose parameter density is 77.06%. These results do not support the hypothesis that noisy updates cause collapse by moving into the principal parameter subspace. Instead, the instability appears to arise mainly from the direction and consistency of the stochastic gradients, rather than from a qualitatively different region of parameter space being updated. Additional cosine-similarity and policy-update analyses in Appendix D.3 show that easy-prompt gradients can also partially cancel the effective update directions contributed by hard prompts, though they give limited gain to the model.

We investigate critic-related sources of gradient noise in Appendix D.2. The results suggest that the critic retains the ability to distinguish prompt difficulty, despite its near-chance ability to judge correct or incorrect sentences within individual trajectories. This may explain why an imperfect critic can still support learning: coarse estimates of prompt difficulty provide useful baseline information, analogous to the group-mean baseline, even when fine-grained credit estimates are unreliable.

We hypothesize that critic errors are consequential for failed rollouts on easy prompts. These trajectories often contain mostly correct reasoning shown in Table 7, interrupted by a few decisive mistakes, making reliable credit assignment dependent on precise error localization. An inaccurate critic may continue to assign high values to unrecoverable errors. Successful rollouts, by contrast, exhibit closer agreement between the terminal outcome and the quality of intermediate reasoning with recoverable mistakes. This asymmetry may make failed easy-prompt rollouts especially sensitive to critic imprecision, contributing to the higher gradient noise observed for easy prompts.

6 Conclusion

We presented SR-PPO, a single-rollout training framework that combines Monte Carlo token-level advantage estimation with Pass@k gradient reweighting. A prefix critic is trained online from terminal outcomes and used both as a baseline and to estimate prompt-level reweighting coefficients. Our mathematical reasoning experiments suggest that this combination supports stable and prolonged policy optimization despite imperfect critic estimates.

Beyond evaluating training performance, we analyzed critic quality, estimation errors, gradient noise, and update behavior. The critic captures coarse differences in prompt difficulty despite providing unreliable within-rollout mistake localization. These findings help explain why critic-based reweighting remains useful: it reduces the contribution of noisy easy-prompt updates and their interference with gradients from harder prompts. Our analysis thus provides an empirical account of both the stabilization mechanism and the residual credit-estimation errors that may limit performance.

Our conclusions are restricted to the DeepScaleR mathematical reasoning dataset, a Qwen3-1.7B base model, and a limited hyperparameter search. Generalization to other models, domains, and long-horizon agentic environments, as well as further investigation on critic training objectives and advantage normalization techniques, remains unexplored. Given computational constraints, we focus on understanding and stabilizing single-rollout learning rather than comprehensively establishing its advantages over multi-rollout training, as investigated by Hou et al. (2026).

Reproducibility Statement

All the code and checkpoints will be released after the double blind period. All the hyperparameters and design choices are already covered in the paper to support reproducibility and transparency.

AI Use Statement

Generative AI tools assist manuscript writing, coding and figure generation. The authors are responsible for the proof, citations, algorithm design, experiment results, interpretation, and final content.

References

Zhipeng Chen, Xiaobo Qin, Youbin Wu, Yue Ling, Qinghao Ye, Wayne Xin Zhao, and Guang Shi. Pass@ k training for adaptively balancing exploration and exploitation of large reasoning models. *arXiv preprint arXiv:2508.10751*, 2025.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025.

Jasper Dekoninck, Nikola Jovanović, Tim Gehrunger, Kári Rögnvaldsson, Ivo Petrov, Chenhao Sun, and Martin Vechev. Beyond benchmarks: Matharena as an evaluation platform for mathematics with LLMs. In *3rd AI for Math Workshop: Toward Self-Evolving Scientific Agents*, 2026a. URL https://openreview.net/forum?id=DmPE4byHuN.

Jasper Dekoninck, Nikola Jovanović, Tim Gehrunger, Kári Rögnvaldsson, Ivo Petrov, Chenhao Sun, and Martin Vechev. Beyond benchmarks: Matharena as an evaluation platform for mathematics with llms. 2026b. URL https://arxiv.org/abs/2605.00674.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 46375–46408. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/420c9f777c0b4f78d515e53cf74d58b2-Paper-Conference.pdf.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Shuo He, Lang Feng, Qi Wei, Xin Cheng, Lei Feng, and Bo An. Hierarchy-of-groups policy optimization for long-horizon agentic tasks. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=T8Dev99qnz.

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. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

Zhenyu Hou, Yujiang Li, Jie Tang, and Yuxiao Dong. Single-rollout asynchronous optimization for agentic reinforcement learning. *arXiv preprint arXiv:2607.07508*, 2026.

Zixia Jia, Jiaqi Li, Yipeng Kang, Yuxuan Wang, Tong Wu, Quansen Wang, Xiaobo Wang, Shuyi Zhang, Junzhe Shen, Qing Li, Siyuan Qi, Yitao Liang, Di He, Zilong Zheng, and Song-Chun Zhu. The AI hippocampus: How far are we from human memory? *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=Sk7pwmLuAY.

Zhewei Kang, Aosong Feng, Sergey Levine, Dawn Song, and Xuandong Zhao. Vimpo: Value-implicit policy optimization for llms, 2026. URL https://arxiv.org/abs/2606.20008.

Ariel Kwiatkowski, Natasha Butt, Ismail Labiad, Julia Kempe, and Yann Ollivier. Likelihood-based reward designs for general llm reasoning. *arXiv preprint arXiv:2602.03979*, 2026.

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.

Yizhi Li, Qingshui Gu, Zhoufutu Wen, Ziniu Li, Tianshun Xing, Shuyue Guo, Tianyu Zheng, Xin Zhou, Xingwei Qu, Wangchunshu Zhou, et al. Treepo: Bridging the gap of policy optimization and efficacy and inference efficiency with heuristic tree-based modeling. *arXiv preprint arXiv:2508.17445*, 2025.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=5PAF7PAY2Y.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Erran Li, Raluca Ada Popa, and Ion Stoica. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. https://huggingface.co/agentica-org/DeepScaleR-1.5B-Preview, 2025.

Sadegh Mahdavi, Muchen Li, Kaiwen Liu, Renjie Liao, and Christos Thrampoulidis. Beyond accuracy: A policy gradient reweighting approach for pass@ k maximization in llms. In *2nd AI for Math Workshop@ ICML 2025*, 2025.

Simon Park, Simran Kaur, and Sanjeev Arora. How does RL post-training induce skill composition? a case study on countdown. In *2nd Workshop on Compositional Learning: Safety, Interpretability, and Agents*, 2026. URL https://openreview.net/forum?id=ncp6JF7XCx.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

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

Shota Takashiro, Soichiro Nishimori, Paavo Parmas, Yongmin Kim, Kohsei Matsutani, Gouki Minegishi, Yusuke Iwasawa, Takeshi Kojima, and Yutaka Matsuo. On advantage estimates for max@ k policy gradients. *arXiv preprint arXiv:2606.06080*, 2026.

Christos Thrampoulidis, Sadegh Mahdavi, and Wenlong Deng. Advantage shaping as surrogate reward maximization: Unifying pass@k policy gradients. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=R1RhBFUk8t.

Hieu Tran, Zonghai Yao, and Hong Yu. Exploiting tree structure for credit assignment in rl training of llms. *arXiv preprint arXiv:2509.18314*, 2025.

Christian Walder and Deep Tejas Karkhanis. Pass@ k policy optimization: Solving harder reinforcement learning problems. *Advances in Neural Information Processing Systems*, 38:152416–152445, 2026.

Junlin Wang, Jue WANG, Ben Athiwaratkun, Ce Zhang, and James Zou. Mixture-of-agents enhances large language model capabilities. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=h0ZfDIrj7T.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce LLMs step-by-step without human annotations. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9426–9439, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.510. URL https://aclanthology.org/2024.acl-long.510/.

Ronald J Willams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256, 1992.

Zijian Wu, Xiangyan Liu, xinyuan zhang, Lingjun Chen, Fanqing Meng, Lingxiao Du, Yiran Zhao, Fanshi Zhang, Yaoqi Ye, Jiawei Wang, Zirui Wang, Jinjie Ni, Yufan Yang, Arvin Xu, and Michael Qizhe Shieh. MCPMark: A benchmark for stress-testing realistic and comprehensive MCP use. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=uobROwBsJm.

Zhiheng Xi, Chenyang Liao, Guanyu Li, Zhihao Zhang, Wenxiang Chen, Binghai Wang, Senjie Jin, Yuhao Zhou, Jian Guan, Wei Wu, Tao Ji, Tao Gui, Qi Zhang, and Xuanjing Huang. Agentprm: Process reward models for llm agents via step-wise promise and progress. In *Proceedings of the ACM Web Conference 2026*, WWW '26, pp. 4184–4195, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400723070. doi: 10.1145/3774904.3792551. URL https://doi.org/10.1145/3774904.3792551.

Chulin Xie, Yangsibo Huang, Chiyuan Zhang, Da Yu, Xinyun Chen, Bill Yuchen Lin, Bo Li, Badih Ghazi, and Ravi Kumar. On memorization of large language models in logical reasoning. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pp. 2742–2785, 2025.

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.

Qianyu Yang, Yang Liu, Jiaqi Li, Jun Bai, Hao Chen, Kaiyuan Chen, Tiliang Duan, Jiayun Dong, Xiaobo Hu, Zixia Jia, et al. $onemillion-bench: How far are language agents from human experts? *arXiv preprint arXiv:2603.07980*, 2026.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2024, 2024.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2025, 2025.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2026, 2026.

Hanqing Zhu, Zhenyu Zhang, Hanxian Huang, DiJia Su, Zechun Liu, Jiawei Zhao, Igor Fedorov, Hamed Pirsiavash, Zhizhou Sha, Jinwon Lee, et al. The path not taken: Rlvr provably learns off the principals. *arXiv preprint arXiv:2511.08567*, 2025.

A PASS@K POLICY GRADIENT THEOREM PROOF

Proof. Using $q_{\pi,k}(s_0) = 1 - (1 - q_\pi(s_0))^k$, the chain rule gives

$$\nabla_\theta J_{\pi,k}(\theta) = \sum_{s_0} d(s_0) w_{\pi,k}(s_0) \nabla_\theta q_\pi(s_0),$$

establishing the first equality.

For each prompt, the likelihood-ratio identity and autoregressive factorization yield

$$\begin{aligned}\nabla_\theta q_\pi(s_0) &= \mathbb{E}_{y_{1:T}\sim\pi_\theta(\cdot|s_0)} \left[Y \nabla_\theta \log \pi_\theta(y_{1:T} \mid s_0) \right] \\ &= \mathbb{E}_{y_{1:T}\sim\pi_\theta(\cdot|s_0)} \left[Y \sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(y_t \mid s_{t-1}) \right].\end{aligned}$$

A fixed pre-action baseline has zero expected contribution conditional on the prefix:

$$\begin{aligned}&\mathbb{E}_{y_t\sim\pi_\theta(\cdot|s_{t-1})} \left[b(s_{t-1}) \nabla_\theta \log \pi_\theta(y_t \mid s_{t-1}) \right] \\ &\quad = b(s_{t-1}) \nabla_\theta \sum_{y_t} \pi_\theta(y_t \mid s_{t-1}) = 0.\end{aligned}$$

For a fixed policy, $w_{\pi,k}(s_0)$ is determined by the prompt and therefore preserves this cancellation. Subtracting the baseline, multiplying by the prompt-level weight, and averaging over prompts gives the second equality. □

B HYPERPARAMTERS

Due to the limited computational resources, most hyperparamters for math reasoning use the default or common setup for LLM post-training as shown in Table 3. But the batch size is differently set up, especially for off-policy and on-policy variants, shown in Table 4.

Actor LR	Critic LR	Clip Ratio High	Entropy Coef.	KL Coef.	Temperature
$1e-6$	$1e-5$	0.25	0.0	0.001	1.2

Table 3: Shared and untuned hyperparamters between all algorithms for math reasoning.

	Train Data BS	Per-Gradient-Update BS	Rollout Number
On-Policy SR-PPO/PPO	128	128	1
Off-Policy GRPO	256	64	4
Off-Policy SR-PPO/PPO	512	128	1

Table 4: Different batch size and rollout number setting for a fair comparison.

The SR-PPO specific hyperparamter is set to be $k = 4$ and $\lambda_{\text{cal}} = 1.0$.

C EVALUATION METRICS

Pass@k Accuracy For $k < 128$, Pass@k is estimated from these samples using the standard unbiased estimator

$$\widehat{\text{Pass@}k} = 1 - \frac{\binom{n-c}{k}}{\binom{n}{k}}, \qquad n = 128,$$

where c denotes the number of correct completions among the n sampled responses.

Noise-to-Signal Ratio To quantify the stability of stochastic policy-gradient estimates, we measure the noise-to-signal ratio (NSR), defined as the gradient variance relative to the squared norm of the mean gradient. For a stochastic gradient estimator $\hat{g}$ with mean $\mu = \mathbb{E}[\hat{g}]$,

$$\mathrm{NSR} = \frac{\mathbb{E}\big[\|\hat{g} - \mu\|_2^2\big]}{\|\mu\|_2^2} = \frac{\operatorname{tr}\operatorname{Cov}(\hat{g})}{\|\mathbb{E}[\hat{g}]\|_2^2}.$$

A lower NSR indicates that stochastic updates are more consistently aligned with the expected optimization direction, whereas a higher NSR reflects stronger rollout-induced noise.

Given R independent gradient replicas $g^{(1)}, \ldots, g^{(R)}$, we estimate the signal magnitude by

$$\widehat{S} = \frac{1}{R(R-1)} \sum_{r \neq s} \left\langle g^{(r)}, g^{(s)} \right\rangle,$$

and the noise magnitude by

$$\widehat{V} = \frac{1}{R} \sum_{r=1}^{R} \|g^{(r)}\|_2^2 - \widehat{S}.$$

The empirical NSR is then

$$\widehat{\mathrm{NSR}} = \frac{\widehat{V}}{\widehat{S}},$$

with $\widehat{\mathrm{NSR}} = \infty$ when $\widehat{S} \leq \epsilon$ due to an unreliable finite-sample signal estimate.

Off-Principal Ratio We measure whether gradient updates concentrate on parameters that are less structurally important to the pretrained checkpoint, following the design in Zhu et al. (2025). For each analyzed weight matrix W_0, we first compute a rank-128 approximation

$$W_0 \approx U_{128} \Sigma_{128} V_{128}^\top,$$

which keeps weights associatd with top 128 singular values.

We define the principal-region mask as the 30% of entries with the largest absolute magnitude in this low-rank reconstruction:

$$M_{\mathrm{princ}} = \mathrm{Top}_{30\%} \left(\left| U_{128} \Sigma_{128} V_{128}^\top \right| \right).$$

We separately define the low-magnitude region as the 30% of entries with the smallest absolute pretrained weights:

$$M_{\mathrm{low}} = \mathrm{Bottom}_{30\%}(|W_0|).$$

The safe region is then $M_{\mathrm{safe}} = M_{\mathrm{princ}}^c \cup M_{\mathrm{low}} = \left(M_{\mathrm{princ}} \cap M_{\mathrm{low}}^c \right)^c$.

Intuitively, this mask excludes parameters that simultaneously have large contribution to the principal low-rank structure and are not small in magnitude. In the analyzed checkpoint, M_{safe} contains 77.06% of the considered parameters.

For each difficulty bin $B \in \{\mathrm{easy}, \mathrm{medium}, \mathrm{hard}\}$, we construct a gradient update

$$\Delta W_B = \eta g_B, \qquad W_B^+ = W_0 + \Delta W_B, \qquad \eta = 0.01.$$

Because training is performed in finite precision, we distinguish the continuous update from the subset of entries whose stored BF16 value actually changes. We define

$$M_{\text{update},B} = \left\{(a,b) : \text{bf16}(W^{+}_{B,ab}) \neq \text{bf16}(W_{0,ab})\right\}.$$

We report the *changed-weight safe-region rate* measures the fraction of BF16-realized parameter changes that occur inside the safe region:

$$R^{\text{change}}_{\text{safe},B} = \frac{|M_{\text{update},B} \cap M_{\text{safe}}|}{|M_{\text{update},B}|}.$$

This metric captures where the update produces an actual change in the stored parameters.

C.1 Critic Quality AUC

AUC measures how well the critic ranks correct sentences above incorrect sentences, without requiring its scores to be calibrated probabilities. This is appropriate because the ground-truth judgments are binary, while the critic outputs arbitrary continuous values.

AUC measures how well the critic ranks correct sentences above incorrect sentences, without requiring its scores to be calibrated probabilities. An AUC of 1.0 indicates perfect ranking, 0.5chance-level ranking, and values below 0.5 indicate that incorrect examples tend to receive higher scores. Ties receive half credit.

For each sentence j in prompt i, we average the critic values over all tokens in that sentence,

$$s_{ij} = \frac{1}{|T_{ij}|} \sum_{t \in T_{ij}} \hat{q}_\phi(s_t),$$

and assign a binary correctness label $y_{ij} \in \{0, 1\}$ from the Codex judgment. We use the critic's raw score, with no clipping or thresholding.

Within each prompt, AUC can be interpreted as the probability that a randomly selected correct sentence receives a higher critic score than a randomly selected incorrect sentence, with ties receiving half credit:

$$\text{AUC}_i = \frac{\sum_{j:y_{ij}=1} \sum_{k:y_{ik}=0} \left[\mathbf{1}(s_{ij} > s_{ik}) + \frac{1}{2}\mathbf{1}(s_{ij} = s_{ik})\right]}{n_i^+ n_i^-},$$

where n_i^+ is the number of positive/correct sentences in prompt i, and n_i^- is the number of negative/incorrect sentences. So $n_i^+ n_i^-$ is simply the total number of correct–incorrect sentence pairs being compared.

Then, we average AUC_i equally across prompts. Prompts containing only correct or only incorrect sentences are excluded because within-prompt AUC is undefined for them, and we report the number of prompts retained. This within-prompt construction focuses the metric on whether the critic can distinguish good from bad reasoning within the same problem.

D Further Empirical Results

D.1 Reasoning and Logic Benchmarks

Knights & Knaves tests logical deduction: given statements from characters who either always tell the truth or always lie, the model must identify each character's type. *GPQA Diamond (GPQA-D)* evaluates scientific knowledge and reasoning through graduate-level multiple-choice questions in biology, chemistry, and physics. *Countdown4* tests arithmetic planning by requiring the model to combine four given numbers using basic arithmetic operations to reach a specified target.

D.2 Critic Quality

We use Codex 5.6-Sol to judge both whether each sentence is correct and whether the solution state remains correct after that sentence. Notice that if the solution is already wrong, then after a correct sentence, the solution state remains negative. We then compute AUC separately for these two binary labels. We also experiment with directly training the critic to predict the Pass@k value function, but observed no significant improvement over a standard Pass@1 critic and therefore do not adopt this design. As shown in Table 5, the resulting per-prompt AUCs are only slightly above 0.5, indicating that the critic is only marginally better than random chance at ranking correct states or sentences above incorrect ones within each prompt's solution.

However, pooling sentences across all 128 prompts yields a higher AUC, suggesting that the critic retains the ability to distinguish quality across prompts despite its limited ability to localize mistakes within individual trajectories. This may explain why an imperfect critic can still support learning: even when fine-grained credit estimates are unreliable, coarse distinctions between easy and hard prompts can remain useful, similar to the group-relative baseline. As shown in Table 6, the critic outputs' magnitudes are indeed proportional to the success rate.

Model	**SR-PPO**	**SR-PPO Pass@k Variant**	**PPO**
Per-Prompt Sentence AUC Mean	0.5362	0.5204	0.4958
Per-Prompt State AUC Mean	0.5693	0.5704	0.5370
All Sentence AUC	0.6735	0.6809	0.6409
All State AUC	0.7385	0.7424	0.7056

Table 5: Critic Quality Score.

Metric	**Easy**	**Medium**	**Hard**
SR-PPO	0.6694 ± 0.2627	0.4498 ± 0.2758	0.2336 ± 0.2499
PPO	0.6699 ± 0.2003	0.4546 ± 0.2216	0.3511 ± 0.2429

Table 6: Averaged token values from the critic for all difficulty bins.

We also observe a substantial drop in the fraction of valid solution states, from 84.99% on easy problems to 48.58% on hard problems. We hypothesize that the higher noise on easy prompts arises because failed rollouts are often mostly valid except for a small number of decisive mistakes, which the critic may fail to localize and may even assign higher values to. In contrast, credit-assignment errors on successful rollouts are less harmful because any intermediate mistakes are ultimately recoverable.

Metric	**Easy**	**Medium**	**Hard**
Sentence Correctness	94.02%	90.22%	80.58%
Prefix State Validity	84.99%	71.28%	48.58%

Table 7: Fraction of correct sentences among different difficulty.

D.3 Gradient Direction Analysis

In Figure 4 Panel (a), each row is the difficulty bin whose data was used for the update; each column is the bin whose correct-answer log-probability is measured. Two structural features stand out.

First, the diagonal is positive everywhere, but its magnitude falls sharply with easiness, meaning that easy prompts have little left to gain. Hard prompts, by contrast, have the most headroom and the largest self-improvement.

Second, and more importantly, the off-diagonal structure is strongly asymmetric. Updating on hard data never meaningfully hurts the other bins ($+0.00198$ on medium, -0.00005 on easy, i.e. neutral).

But updating on medium or easy data actively degrades hard-prompt correctness, by -0.01229 and -0.00566 respectively.

Panel (b) shows that the cosine similarities supply the mechanism. Hard-bin gradients are strongly anti-aligned with both medium (-0.559) and easy (-0.567), while medium and easy are mutually aligned ($+0.521$). Easier prompts therefore form a coherent group pulling parameters in one direction, and that direction is close to the opposite of what hard prompts require.

Putting the two together. The negative cosines explain why easy and medium updates push hard-prompt likelihood down, though easy answers sit near the ceiling and have little probability to gain.

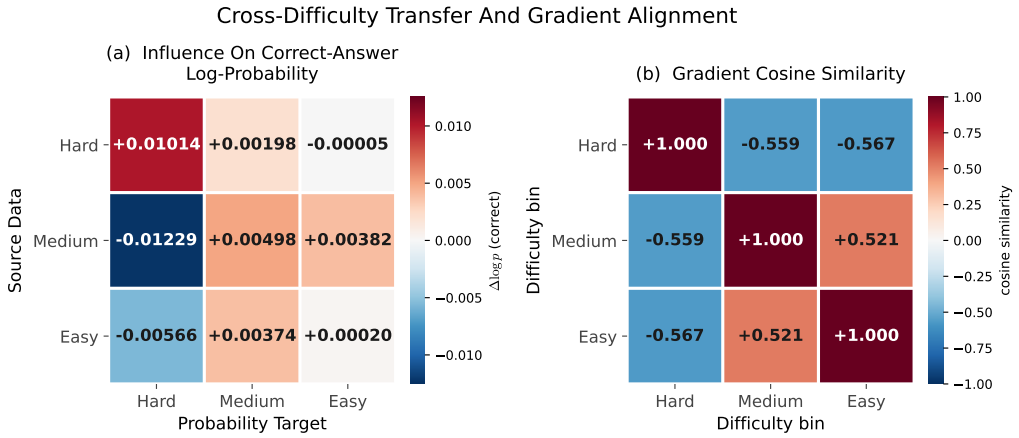


Figure 4: **Gradient direction analysis.** Panel (a): cross-difficulty influence. Each row is the difficulty bin whose data was used for the update; each column is the bin whose correct-answer log-probability is measured. Panel (b): gradient alignment. The cosine similarities supply the mechanism. Hard-bin gradients are strongly anti-aligned with both medium (-0.559) and easy (-0.567).