RETHINKING CRITIC LEARNING IN PPO: UNDERSTANDING AND MITIGATING VALUE FLATTENING

Anonymous authors
Paper under double-blind review

ABSTRACT

In reinforcement learning for large language models, Proximal Policy Optimization (PPO) commonly uses a critic to estimate state values and reduce the variance of policy updates. However, we uncover a systematic failure mode in PPO critics, which we call **value flattening**: state values, estimated from multiple Monte Carlo continuations, change sharply across intermediate states while critic predictions remain comparatively flat. We further observe this phenomenon in a controlled FrozenLake environment and find that it becomes more pronounced as the state space grows. Our theoretical and empirical analyses relate value flattening to an implicit variance penalty in the sampled loss and redundant updates from temporally correlated states with similar gradients. Motivated by these findings, we introduce SParse Proximal Policy Optimization ($\mathrm{SP}^3\mathrm{O}$), which applies the value loss to only a few well-separated states in each response to mitigate both effects. Experiments on Qwen3-Base show that $\mathrm{SP}^3\mathrm{O}$ with only three states supervised per response can mitigate value flattening and consistently improve the learned policy across model sizes and evaluation suites. Together, our results identify value flattening as an important yet overlooked failure mode of critic learning in standard PPO and show that a simple sparse supervision strategy can mitigate it.

1 INTRODUCTION

Reinforcement learning enables large language models to tackle increasingly challenging reasoning tasks that unfold over long trajectories (Kimi Team et al., 2026; Li et al., 2026b; Chen et al., 2026a). Such long-horizon tasks pose a fundamental challenge for policy optimization, which requires fine-grained credit assignment over intermediate decisions (Hou et al., 2026; Kazemnejad et al., 2025; Guo et al., 2025; Wang et al., 2026; Gong et al., 2026). Proximal Policy Optimization (PPO) offers a natural framework for meeting this demand by using a critic to estimate state values throughout each trajectory and construct token-level advantages (Schulman et al., 2017; 2015b; Yuan et al., 2025; Yue et al., 2025; Qi et al., 2026). This, in turn, places a key requirement on the critic: it must capture meaningful changes in expected success (i.e., the state value) across the trajectory.

However, we find that the critic in standard PPO fails to capture substantial changes in state values across steps within a response. We refer to this phenomenon as **value flattening**. For each intermediate state, we estimate its state value by averaging terminal rewards from multiple independent continuations sampled from the same policy, obtaining Monte Carlo estimates of state values (MC values) (Schulman et al., 2015a; Wang et al., 2023; Kazemnejad et al., 2025). We use these estimates as diagnostic references for evaluating critic predictions along individual trajectories. Figure 1 illustrates this mismatch by comparing critic predictions with MC values. Across these trajectories, MC values often exhibit sharp local transitions, whereas critic predictions remain comparatively flat. In some cases, the critic prediction changes in the opposite direction from the MC values. These examples also show that critic predictions remain relatively insensitive to local variation in MC values across training checkpoints and in both correct and incorrect responses, indicating that value flattening is a systematic property of the learned critic.

To further characterize value flattening, we study a stochastic FrozenLake environment. In this gridworld, an agent navigates from a start state to a goal while avoiding holes. We observe that as the maze size increases, critic predictions become progressively smoother and less accurate (Figure 2(b,c)). Taken together, our observations in LLMs and FrozenLake suggest that value

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.

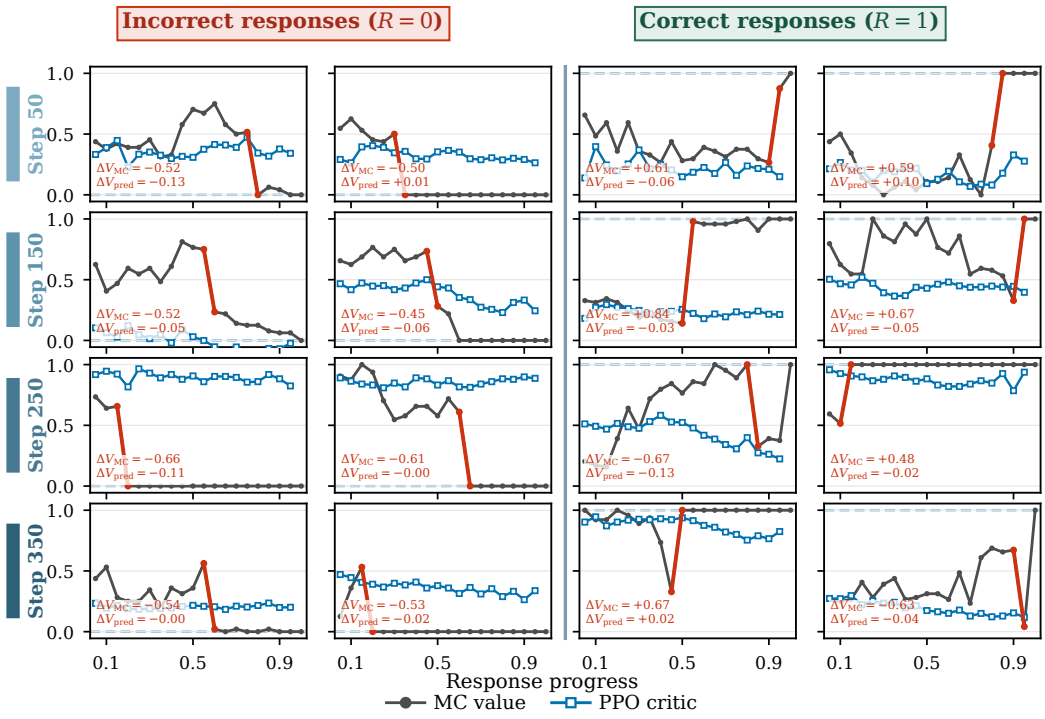

Figure 1: Value flattening across training in a Qwen3-4B-Base PPO run on DAPO-Math-17k.

flattening is a systematic critic failure mode that may become more pronounced as the state space grows, motivating us to investigate its underlying causes.

To understand these phenomena, we analyze the critic training objective and the temporal dependence in its supervision, identifying two factors that can contribute to value flattening (Section 4.2): **(1) Implicit Variance Penalty in the Sampled Loss:** Under the common setup in PPO training with terminal-only rewards (Yuan et al., 2025; Yue et al., 2025; Hu et al., 2025), the critic's mean squared error (MSE) loss is applied at every token position in a response. Our loss decomposition shows that this objective contains a term penalizing variation among value predictions within a response. This term creates pressure toward value flattening during finite-batch optimization. **(2) Redundant Updates from Temporal Correlation:** LLM states consist of the tokens generated so far, so neighboring states differ by only one token and are highly temporally correlated. We find that nearby states have similar representations and gradients in the critic, with gradient similarity decreasing as the distance between token positions increases. Dense token-level supervision therefore aggregates many similar updates from neighboring states, which can make their predicted values more similar.

Motivated by these findings, we propose SParse Proximal Policy Optimization (SP^3O). Supervising fewer, more widely separated positions restricts the implicit variance penalty to their predicted values and is designed to reduce redundant updates from temporally correlated states. Experiments show that SP^3O mitigates value flattening and consistently improves actor performance across Qwen3-Base on mathematical and general reasoning benchmarks. Our contributions are threefold:

- We identify value flattening, a systematic mismatch in which state values estimated from multiple Monte Carlo continuations can change sharply within individual responses while PPO critic predictions remain comparatively flat.
- We relate value flattening to two factors: the implicit variance penalty in the sampled critic MSE loss, which penalizes value differences across token positions, and redundant updates from temporally correlated states with similar gradients.
- We introduce SP^3O, which supervises the critic at a few well-separated states to mitigate the effects of the implicit variance penalty and redundant neighboring updates. SP^3O mitigates value flattening and improves policy performance across model scales and evaluation suites.

2 Related Work

Critic-Free and Critic-Based RL. Critic-free methods such as RLOO (Ahmadian et al., 2024), GRPO (Shao et al., 2024), and DAPO (Yu et al., 2025) avoid training a value model, but their advantages are derived from complete responses and offer little direct distinction among states within the same response. Recent work such as VIMPO derives an implicit value function to recover finer-grained credit without a separate critic (Kang et al., 2026). Critic-based methods such as PPO instead learn a value function and use its predictions at each generation step to construct token-level advantages (Schulman et al., 2017). Building on PPO, recent methods improve critic initialization and optimization for reasoning or adapt critic-based learning to asynchronous and structured trajectories (Yuan et al., 2025; Yue et al., 2025; Hou et al., 2026; Li et al., 2026c; He et al., 2026; Chen et al., 2025b; Luo et al., 2026; Qi et al., 2026), but provide little analysis of whether their critic predictions capture value changes within individual responses. We study this behavior and identify value flattening: PPO critics retain differences across responses but fail to track value changes among states within the same response.

Fine-Grained Credit Assignment and Value Estimation. Prior work obtains intermediate credit through outcomes organized by segments or trees and process supervision (Guo et al., 2025; Hou et al., 2025; Tran et al., 2025; Ielanskyi et al., 2026; Lightman et al., 2023). These signals provide finer feedback but do not directly estimate the policy-conditioned state value: the expected terminal return when the current policy continues from an intermediate state. VinePPO estimates this quantity with auxiliary continuations and improves credit assignment, but fine-grained estimation requires additional rollouts from each evaluated state and can become costly as responses lengthen (Kazemnejad et al., 2025; Wang et al., 2026; Gong et al., 2026; Shan et al., 2026). Hista estimates state values by weighting returns from existing rollouts according to hidden-state similarity (Chen et al., 2026b). We build on this line of work by evaluating both response-level discrimination and the ability of PPO critics to track policy-conditioned state-value changes within individual responses across training stages and supervision densities.

3 Preliminaries

3.1 Proximal Policy Optimization

Given a prompt x, a policy π_θ generates a response $y = (y_1, \ldots, y_T)$ and thereby a trajectory τ, where the state and action at step t are $s_t = (x, y_{<t})$ and $a_t = y_t$. The return from s_t is $G_t = \sum_{k=t}^{T} \gamma^{k-t} r_k$. PPO estimates advantages from trajectories sampled by $\pi_{\theta_{\text{old}}}$ and maximizes the clipped surrogate objective (Schulman et al., 2017)

$$\mathcal{L}_{\text{PG}}(\theta) = \mathbb{E}_t\Big[\min\Big(\rho_t(\theta)\widehat{A}_t, \text{clip}(\rho_t(\theta), 1-\epsilon, 1+\epsilon)\,\widehat{A}_t\Big)\Big], \tag{1}$$

where $\rho_t(\theta) = \pi_\theta(a_t \mid s_t)/\pi_{\theta_{\text{old}}}(a_t \mid s_t)$ and $\widehat{A}_t$ is the estimated advantage.

PPO uses a critic $V_\phi(s_t)$ to approximate the policy-conditioned value $V^\pi(s_t) = \mathbb{E}_\pi[G_t \mid s_t]$. Given the advantage $\widehat{A}_t$ computed by generalized advantage estimation (GAE) (Schulman et al., 2015b), the critic regression target is

$$\widehat{G}_t = \widehat{A}_t + V_{\phi_{\text{old}}}(s_t). \tag{2}$$

The full GAE definition is given in Appendix A.1. For a rollout batch $\mathcal{B}$, the standard critic update is:

$$\mathcal{L}_V(\phi) = \frac{1}{\sum_{\tau\in\mathcal{B}} T_\tau} \sum_{\tau\in\mathcal{B}} \sum_{t=1}^{T_\tau} \Big(V_\phi(s_t) - \widehat{G}_t\Big)^2, \qquad \phi \leftarrow \phi - \eta_V \nabla_\phi \mathcal{L}_V(\phi). \tag{3}$$

For the experiments in this work, the KL coefficient is zero and the task reward is terminal. Let $R(\tau)$ denote the realized terminal task reward of trajectory τ, which is binary in our experiments. Thus, $r_t = 0$ for $t < T$ and $r_T = R(\tau)$; with $\gamma = \lambda = 1$,

$$\widehat{G}_t = G_t = R(\tau), \qquad t = 1, \ldots, T.$$

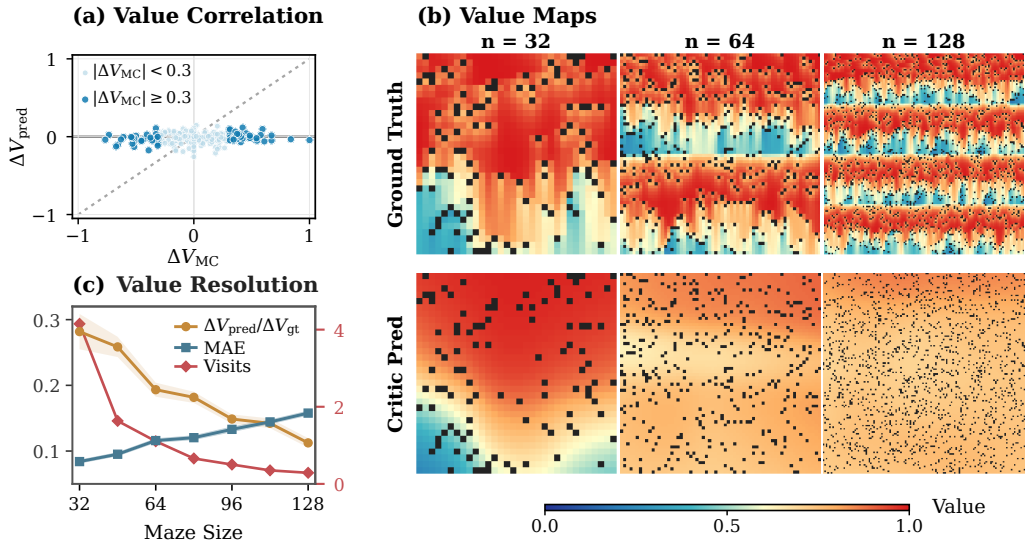

Figure 2: Value Flattening in LLM Reasoning and stochastic FrozenLake. **(a)** Adjacent MC and critic value changes: predicted changes remain concentrated near zero even when MC values change sharply. **(b)** Ground-truth and critic value maps as the FrozenLake maze size n increases under an otherwise fixed training configuration. **(c)** The corresponding value-resolution statistics.

3.2 Monte Carlo Estimation of State Value

A sampled return G_t is a single-rollout Monte Carlo estimate of $V^\pi(s_t)$. We obtain a lower-variance diagnostic estimate by independently sampling K continuations $\tau_t^{(1)}, \ldots, \tau_t^{(K)}$ from the same policy conditioned on a fixed state s_t:

$$\widehat{V}^\pi_{\mathrm{MC}}(s_t) = \frac{1}{K} \sum_{k=1}^{K} G_t^{(k)}, \tag{4}$$

where $\tau_t^{(k)} \sim \pi(\cdot \mid s_t)$ and $G_t^{(k)} = R(\tau_t^{(k)})$ is its realized return in our terminal-reward setting. This sample average estimates the theoretical state value $V^\pi(s_t) = \mathbb{E}_\pi[G_t \mid s_t]$. For the terminal binary-reward setting used here, it is the empirical success rate of the sampled continuations. We use it as a diagnostic reference to distinguish fitting the single-rollout target $\widehat{G}_t$ from agreement with the policy-conditioned state value.

4 Understanding and Mitigating Value Flattening

We first show that value flattening occurs in both LLM reasoning and a controlled Markov decision process, and that it becomes more pronounced as the state space grows. We then examine how shared terminal-return targets create a within-response variance term in the sampled critic loss and how dense supervision produces redundant updates from temporally correlated states. This analysis motivates sparse critic supervision. Unless otherwise stated, the LLM analysis uses Qwen3-4B-Base trained on DAPO-Math-17k. Full training and figure-specific settings are provided in Appendix A.1.

4.1 Value Flattening in Critic Learning

Value Flattening in LLM Reasoning. We characterize value flattening directly through critic predictions. Across training checkpoints and for both correct and incorrect responses, MC values can change sharply between adjacent reasoning states while the corresponding critic profiles remain comparatively flat (Figure 1). The adjacent-change analysis in Figure 2(a) makes this mismatch explicit: MC value changes span a broad range, whereas critic changes remain concentrated near zero instead of following the diagonal. The critic therefore often fails to capture both the magnitude and the direction of local value changes. Consequently, reasoning states with substantially different MC

values can receive similar critic predictions within the same response. Appendix A.2 formalizes this mismatch through an error decomposition.

Value Flattening with State-Space Growth. To examine whether value flattening extends beyond LLMs and how it scales with state-space size, we study stochastic FrozenLake (Brockman et al., 2016), where an agent navigates from a start state to a goal while avoiding holes. Each episode has a binary return, equal to 1 upon reaching the goal and 0 otherwise, so the state value of a grid cell is the probability of eventually reaching the goal from that state. This setting provides a controlled experiment in which we hold the training configuration fixed and increase the maze size n to expand the state space. As shown in Figure 2(b), critic predictions across grid cells become progressively smoother as the maze grows. The local-contrast and error metrics in panel (c) likewise show smoother value predictions and weaker agreement with the ground truth. Thus, value flattening becomes more pronounced as the state space grows.

4.2 Why Does Value Flattening Occur?

We examine two potential contributors to finite-batch value flattening: the within-response variance penalty in the sampled loss and redundant critic updates.

Implicit Variance Penalty in the Sampled Loss. Under terminal-only rewards with $\gamma = \lambda = 1$, a common setting in LLM reasoning PPO (Hu et al., 2025; Wang et al., 2026), standard PPO uses the same sampled terminal return as the target at every state in a response.

For a response of length T, let $v_t = V_\phi(s_t)$ and $\bar{v} = T^{-1}\sum_{t=1}^{T} v_t$. Then

$$\frac{1}{T}\sum_{t=1}^{T}(v_t - R)^2 = (\bar{v} - R)^2 + \frac{1}{T}\sum_{t=1}^{T}(v_t - \bar{v})^2. \tag{5}$$

The first term fits the response mean to R; the second penalizes variation because the sampled target profile is constant. Although the population MSE has $V^\pi(s)$ as its pointwise optimum, recovering state-value variation requires averaging and generalization across trajectories. Finite batches provide limited independent outcomes for such averaging, so the variance term exerts pressure toward value flattening during optimization.

Redundant Updates from Temporal Correlation. Classical RL often operates on compact Markov states that summarize the information needed for future decisions. In LLM post-training, by contrast, a state contains the tokens generated so far, so adjacent states differ by only one token and overlap in nearly their entire input. Because adjacent states are temporally correlated and share most of their tokens, supervising all of them provides less diverse training signals than their number suggests (Mnih et al., 2015). This similarity is also reflected in the critic hidden representations, whose trajectory in Figure 3(a) occupies a compact region.

Representation similarity connects directly to update similarity. Consider a value head $V_\phi(s_t) = w^\top h_t$, where h_t is the critic representation. The gradient from position t is $g_t^{(w)} = 2\big(V_\phi(s_t) - R\big)h_t$. Thus, positions with similar representations and residuals of the same sign produce aligned gradients. The measurements in Figure 3(b) show that hidden-state alignment, gradient alignment, and update energy remain high throughout training. Within each response, gradient similarity decreases as the distance between their token positions increases, consistent with stronger temporal correlation between nearby states. These results indicate that dense supervision produces many redundant updates from neighboring states.

4.3 SP^{3}O: Sparse Critic Supervision

Sparse supervision is designed to mitigate the two problems above through selection and spacing. Selecting fewer positions restricts the per-response variance penalty to those predictions instead of applying it at every token position. Spacing the selected positions apart is designed to reduce the accumulation of aligned gradients from nearby states that share most of their tokens.

We instantiate this intervention as SParse Proximal Policy Optimization (SP^{3}O). The actor objective, rollout procedure, and return targets remain unchanged; only the states receiving the critic loss are changed. SP^{3}O applies the value loss only at a small set of well-separated states. The critic still

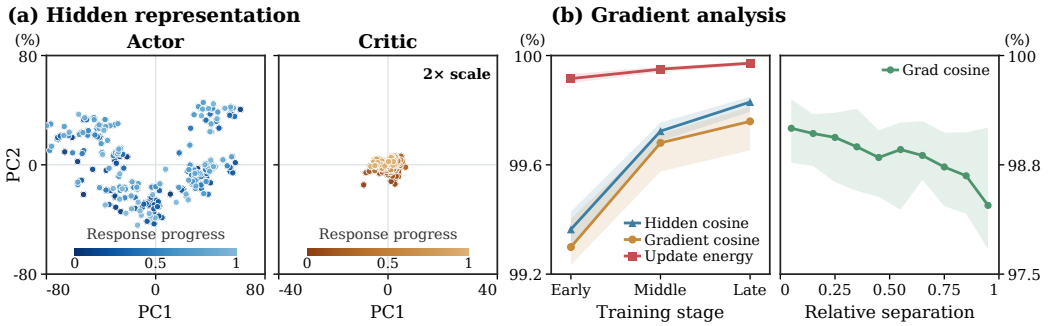

Figure 3: Representation and gradient analysis in Qwen3-4B-Base PPO critic supervision. **(a)** Actor and critic hidden representations for the same response, centered and normalized by mean hidden norm; the critic view uses a 2× scale. **(b)** Hidden-representation alignment, gradient alignment, and update energy remain high across PPO training stages (left), while the full critic gradient cosine decreases with relative token-position separation (right).

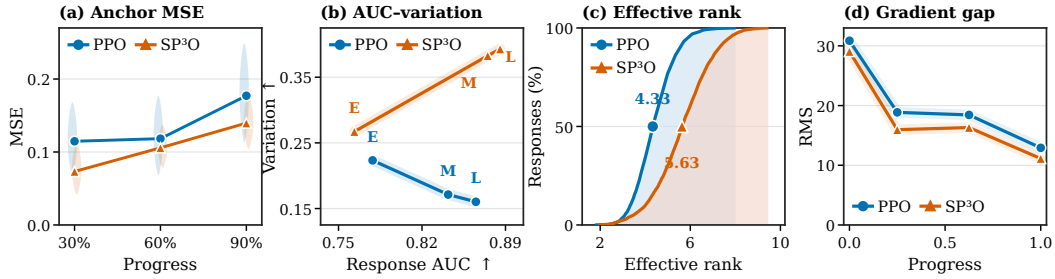

Figure 4: Effects of sparse supervision on critic prediction and optimization. **(a)** Response-centered value MSE with 95% bootstrap intervals; **(b)** response AUC versus within-response value variation; **(c)** effective-rank distributions; **(d)** value-head gradient RMS difference. Definitions are in Appendix A.1.

produces values at every generation state. Let $\mathcal{I}(\tau)$ denote the supervised states in trajectory τ. The sparse critic objective is

$$\mathcal{L}_V^{\mathrm{SP^3O}}(\phi) = \frac{1}{\sum_{\tau\in\mathcal{B}} |\mathcal{I}(\tau)|} \sum_{\tau\in\mathcal{B}} \sum_{t\in\mathcal{I}(\tau)} \left(V_\phi(s_t) - \widehat{G}_t\right)^2. \tag{6}$$

Critic Prediction Effects. We compare both critics with policy-conditioned MC values obtained from repeated continuations at intermediate states. The aggregate comparison in Figure 4(a) shows that SP^{3}O reduces response-centered MSE at different points in the response. These results indicate that sparse supervision better preserves within-response value variation and improves within-response value resolution.

Critic Optimization Effects. Figure 4 examines how sparse supervision affects critic representations and optimization. Panel (b) compares response-level outcome discrimination with within-response critic-prediction variation across the early (**E**), middle (**M**), and late (**L**) training stages. PPO gains discrimination while losing within-response variation, whereas SP^{3}O improves both. In panel (c), SP^{3}O shifts the effective rank of response-centered hidden states upward, increasing the median from 4.33 to 5.63 and indicating richer representations of state differences. Panel (d) shows a smaller RMS difference between value-head gradients induced by terminal-return targets and MC values throughout the response. Together, these observations are consistent with sparse supervision preserving richer hidden representations and reducing redundant or distorted critic updates, without sacrificing response-level discrimination.

Table 1: In-domain mathematical-reasoning accuracy (%). The final column averages the seven listed tasks. Our method is shaded in light green, and bold denotes the best result within each model block.

Model	Method	Mathematical Reasoning (avg@32)							
		AIME24	AIME25	AIME26	AMC23	MATH500	Minerva	Olympiad	Avg.
Qwen3-4B-Base	Base	4.58	3.75	4.38	25.08	42.78	22.71	22.35	17.95
	PPO	17.50	19.90	14.69	61.80	70.15	42.82	36.35	37.60
	GRPO	17.19	16.19	10.10	63.83	78.21	45.71	43.62	39.26
	HISTA	16.15	10.10	10.52	65.16	79.75	46.62	43.70	38.86
	SP^{3}O	**23.02**	**23.54**	**22.08**	**69.92**	**83.79**	**47.93**	**48.68**	**45.57**
Qwen3-8B-Base	Base	7.40	7.92	5.94	37.66	54.08	24.69	27.36	23.58
	PPO	30.21	25.10	23.33	72.19	86.33	48.81	53.56	48.50
	GRPO	28.50	22.04	22.70	73.82	85.26	**52.06**	50.98	47.91
	HISTA	28.33	23.33	25.00	74.38	83.95	51.29	49.07	47.91
	SP^{3}O	**33.91**	**28.02**	**27.39**	**75.00**	**87.33**	47.40	**54.50**	**50.51**

Actor Optimization Effects. The critic is intended to stabilize PPO updates by replacing raw outcome signals with advantage estimates that reduce policy-gradient variance. Its within-response resolution can therefore affect both the direction and the stability of actor optimization. In the online runs, SP^{3}O has smaller within-iteration update changes over most of training than PPO (Figure 5(a)). This pattern is consistent with smoother actor optimization while retaining the benefits of a learned critic for token-level advantage estimation.

5 Experiments

5.1 Experimental Setup

Models and baselines. We evaluate the proposed sparse critic supervision on Qwen3-4B-Base and Qwen3-8B-Base (Yang et al., 2025). We train both models on DAPO-Math-17k (Yu et al., 2025). For each model, we report the initial checkpoint as a reference and compare our method against standard PPO, GRPO, and Hista (Chen et al., 2026b). Our method, SParse Proximal Policy Optimization (SP^{3}O), combines sparse critic supervision with explicit late-tail coverage. Unless otherwise stated, SP^{3}O supervises response-relative states at 0.3, 0.6, and 0.9, adding a 0.95 state for responses of at least 6K tokens. We additionally evaluate Qwen3-14B-Base trained on Polaris-V2-RL-14K (POLARIS-Project, n.d.) and Qwen3.5-9B trained on scientific coding data generated by SciWalker (Li et al., 2026a) to assess generalization across model sizes, training datasets, and task domains beyond mathematical reasoning.

Benchmarks and evaluation. For mathematical and general reasoning, we use the 24K evaluation setting reported in the result sheets. For coding, we use a 32K token generation limit. **Mathematical reasoning (avg@32).** We use AIME24, AIME25, AIME26, and AMC23 (Mathematical Association of America, n.d.), MATH500 (Lightman et al., 2023), Minerva (Lewkowycz et al., 2022), and OlympiadBench (He et al., 2024). **General reasoning (OOD; avg@4).** We use ARC-C (Clark et al., 2018), MMLU-Pro (Wang et al., 2024), GPQA (Rein et al., 2024), AGIEval-English (Zhong et al., 2024), BigBenchHard (Suzgun et al., 2023), and ZebraLogic-Grid (Lin et al., 2025; WildEval, n.d.). The latter three benchmarks are scored using xVerify (Chen et al., 2025a). **Coding.** We use SciCode (Tian et al., 2024), HumanEval (Chen et al., 2021), HumanEvalFix (Python and Rust) (Muennighoff et al., 2023), and LiveCodeBench (execution v2 and code generation v6) (Jain et al., 2024). We report three-run averages of SciCode subproblem accuracy and pass@1 for the other coding benchmarks, with one response per task in each run.

5.2 Main Results

Tables 1 and 2 show that SP^{3}O achieves higher average accuracy than standard PPO, GRPO, and Hista across both model sizes and evaluation suites. The gains over PPO reach 7.97 percentage points

Table 2: Out-of-distribution evaluation accuracy (%). Datasets marked with † are scored by the `gpt-oss-120b` verifier through xverify. The final column is the unweighted mean over all six tasks. Our method is shaded in light green, and bold denotes the best result within each model block.

Model	Method	General Reasoning (avg@4)						
		ARC-C	MMLU-Pro	GPQA	AGIEval†	BBH†	ZebraLogic†	Avg.
Qwen3-4B-Base	Base	34.98	16.17	14.02	27.93	21.11	1.35	19.26
	PPO	89.19	54.39	34.85	65.87	57.50	9.90	51.95
	GRPO	90.96	56.87	38.89	68.18	71.17	12.58	56.44
	HISTA	90.19	53.18	**39.65**	68.31	71.80	12.70	55.97
	SP^{3}O	**91.02**	**61.75**	38.89	**71.44**	**73.35**	**19.20**	**59.28**
Qwen3-8B-Base	Base	61.82	36.22	27.02	48.59	50.71	4.90	38.21
	PPO	**93.84**	64.19	47.22	74.52	79.27	27.25	64.38
	GRPO	93.04	**66.27**	49.49	75.15	78.10	27.40	64.91
	HISTA	93.22	63.08	48.48	72.74	76.98	19.70	62.37
	SP^{3}O	93.13	65.83	**49.94**	**76.95**	**80.89**	**31.45**	**66.37**

on in-domain mathematical reasoning and 7.33 percentage points on out-of-distribution reasoning, with positive improvements also observed for Qwen3-8B-Base. These results indicate that mitigating critic value flattening through sparse supervision leads to better policy learning rather than merely improving critic-side diagnostics.

Online learning dynamics. Figure 5 compares the training dynamics of PPO and SP^{3}O on Qwen3-4B-Base. As shown in (a), SP^{3}O exhibits smaller and less variable within-iteration actor updates, indicating more stable training. It also maintains higher validation accuracy and rollout reward than PPO after the early training stage, as shown in (b) and (c). Meanwhile, SP^{3}O produces longer responses, whereas response lengths under PPO increase more gradually and remain shorter, as shown in (d). Overall, SP^{3}O achieves stronger performance while maintaining more stable policy updates.

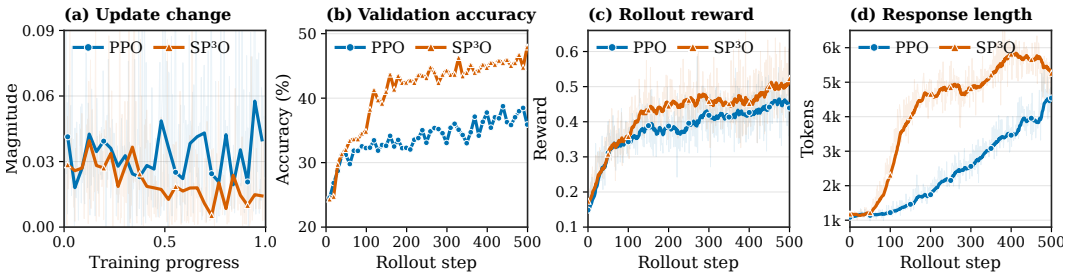

Figure 5: Online learning dynamics and within-iteration actor updates on Qwen3-4B-Base. **(a)** Within-iteration update change. **(b)** Validation accuracy. **(c)** Rollout reward. **(d)** Response length.

5.3 Ablations and Generalization

Sparse-supervision ablations. We examine supervision density, anchor placement, and late-tail coverage on Qwen3-4B-Base. For the density ablation, we vary the number of supervised states over $K \in \{3, 4, 8, 16, 64\}$ and compare with standard PPO (Figure 6). Sparse settings achieve higher performance and tend toward the PPO baseline as K increases. Adding anchors trains more states toward the same terminal return and can introduce similar updates from nearby states, potentially weakening distinctions between their predicted values.

At K = 3, both fixed-position schemes outperform random placement (Table 3 and Figure 10), supporting the use of well-spaced anchors to cover the trajectory while limiting redundant updates on temporally correlated states. Adding the final tail anchor while keeping the other anchors unchanged further improves performance from 44.10% to 45.57% and reduces repetition from 18.33% to 1.12% (Appendix Table 6). Together, these results favor sparse, well-spaced supervision with explicit tail coverage, rather than simply supervising more states.

Table 3: Anchor-placement ablation on Qwen3-4B-Base.

Main anchors	Acc. (%)
PPO baseline	37.60
Random	36.59
0.2/0.5/0.8	44.65
0.3/0.6/0.9	**45.57**

Generalization across model sizes and domains. We further examine whether SP^{3}O's gains extend to larger models and different domains. On the more challenging Polaris-V2-RL-14K mathematical training set, SP^{3}O with Qwen3-14B-Base exceeds PPO by 4.12 and 4.31 percentage points and GRPO by 3.80 and 5.18 percentage points on mathematical and OOD reasoning, respectively (Figure 7). Compared with the main DAPO-Math-17k experiments, this comparison uses a larger model and different mathematical training data while retaining the OOD gain.

To examine transfer beyond mathematical reasoning, we also train Qwen3.5-9B on scientific coding data generated by SciWalker (Li et al., 2026a). The coding evaluations cover program generation, repair, and execution. Relative to PPO, SP^{3}O raises SciCode subproblem accuracy by 1.97 percentage points (from 32.64 to 34.61) and the six-benchmark average by 2.38 percentage points (from 70.81 to 73.19), with gains on five of the six benchmarks (Figure 8). Together, these results suggest that SP^{3}O remains effective across model sizes and domains.

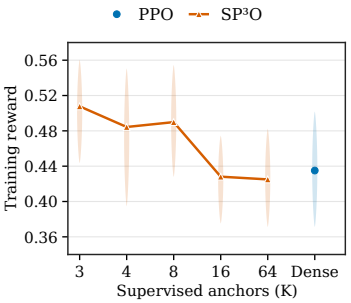

Figure 6: Effect of critic supervision density on performance.

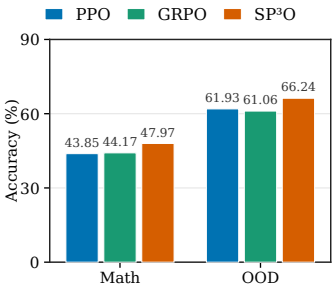

Figure 7: Evaluation results of Qwen3-14B-Base trained on Polaris-V2-RL-14K.

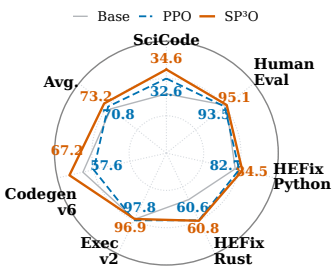

Figure 8: Evaluation results of Qwen3.5-9B trained on SciWalker data.

6 Conclusion

We identify value flattening as a systematic failure mode of critics for large language model reasoning, in which critic predictions fail to capture policy-conditioned state value changes within individual responses. The same pattern appears in controlled stochastic FrozenLake experiments and becomes more pronounced as the state space grows. Our analyses relate this behavior to two factors: an implicit variance penalty in the sampled loss and redundant updates from temporally correlated states with similar gradients. Motivated by these findings, SP^{3}O applies the critic loss at a few well-separated states to mitigate the effects of both factors. Experiments on Qwen3-4B-Base and Qwen3-8B-Base show that SP^{3}O mitigates value flattening and improves actor performance across mathematical and out-of-distribution reasoning benchmarks. Additional results with Qwen3-14B-Base on mathematical reasoning and Qwen3.5-9B on scientific coding suggest that SP^{3}O generalizes across model sizes and domains. These results establish value flattening as an important yet overlooked problem in critic learning.

Reproducibility statement

The critic objective and anchor-selection rule are specified in Section 4.3. Model, data, training, and evaluation settings are given in Section 5.1 and Appendix A.1; additional derivations and ablations appear in Appendices A.2 and A.4. An anonymized code snapshot with reproduction instructions is available at https://anonymous.4open.science/r/SP3O_ICLR-5C74/.

AI use statement

We used generative AI tools to assist with language polishing, including grammar and punctuation, as well as manuscript layout and formatting.

References

Arash Ahmadian, Chris Cremer, Matthias Galle, et al. Back to basics: Revisiting REINFORCE style optimization for learning from human feedback in LLMs. *arXiv preprint arXiv:2402.14740*, 2024. URL https://arxiv.org/abs/2402.14740.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym. *arXiv preprint arXiv:1606.01540*, 2016. URL https://arxiv.org/abs/1606.01540.

Ding Chen, Qingchen Yu, Pengyuan Wang, Mengting Hu, Wentao Zhang, Zhengren Wang, Bo Tang, Feiyu Xiong, Xinchi Li, Chao Wang, Minchuan Yang, and Zhiyu Li. xVerify: Efficient answer verifier for reasoning model evaluations. *arXiv preprint arXiv:2504.10481*, 2025a. URL https://arxiv.org/abs/2504.10481.

Jiacheng Chen, Qianjia Cheng, Fangchen Yu, Haiyuan Wan, Yuchen Zhang, Shenghe Zheng, Junchi Yao, Qingyang Zhang, Haonan He, Yun Luo, Yufeng Zhao, Futing Wang, Li Sheng, Chengxing Xie, Yuxin Zuo, Yizhuo Li, Wenxauan Zeng, Yulun Wu, Rui Huang, Dongzhan Zhou, Kai Chen, Yu Qiao, Lei Bai, Yu Cheng, Ning Ding, Bowen Zhou, Peng Ye, and Ganqu Cui. P1: Mastering physics olympiads with reinforcement learning, 2025b. URL https://arxiv.org/abs/2511.13612.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. URL https://arxiv.org/abs/2107.03374.

Qiguang Chen, Chengyu Luan, Jiajun Wu, Qiming Yu, Yi Yang, Yizhuo Li, Jingqi Tong, Xiachong Feng, Libo Qin, and Wanxiang Che. OMIBench: Benchmarking olympiad-level multi-image reasoning in large vision-language model. *arXiv preprint arXiv:2604.20806*, 2026a. URL https://arxiv.org/abs/2604.20806.

Zizhe Chen, Jiqian Dong, Yizhou Tian, Garry Yang, Yongqiang Chen, Zhitang Chen, and James Cheng. Hista and Numca: Estimate state value effectively for LLM reinforcement learning. *arXiv preprint arXiv:2605.29782*, 2026b. URL https://arxiv.org/abs/2605.29782.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018. URL https://arxiv.org/abs/1803.05457.

Xue Gong, Qi Yi, Ziyuan Nan, et al. Segmental advantage estimation: Enhancing PPO for long-context LLM training. *arXiv preprint arXiv:2601.07320*, 2026. URL https://arxiv.org/abs/2601.07320.

Yiran Guo, Lijie Xu, Jie Liu, Dan Ye, and Shuang Qiu. Segment policy optimization: Effective segment-level credit assignment in RL for large language models. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2505.23564.

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. URL https://arxiv.org/abs/2402.14008.

Haonan He, Haodi Lei, Yun Luo, Haoran Zhang, Shunkai Zhang, Yizhuo Li, Shengji Tang, Zhilin Wang, Runzhe Zhan, Lei Bai, Ganqu Cui, Fangchen Yu, Yafu Li, Peng Ye, Ning Ding, and Yu Cheng. SimpleOPD: Simple tokenizer-agnostic on-policy distillation for long-context reasoning, 2026. URL https://arxiv.org/abs/2608.14277.

Zhenyu Hou, Ziniu Hu, Yujiang Li, et al. TreeRL: LLM reinforcement learning with on-policy tree search. In *Annual Meeting of the Association for Computational Linguistics*, 2025. URL https://arxiv.org/abs/2506.11902.

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

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-Reasoner-Zero: An open source approach to scaling up reinforcement learning on the base model. *arXiv preprint arXiv:2503.24290*, 2025. URL https://arxiv.org/abs/2503.24290.

Mykyta Ielanskyi, Kajetan Schweighofer, Lukas Aichberger, and Sepp Hochreiter. RREDCoT: Segment-level reward redistribution for reasoning models. *arXiv preprint arXiv:2606.06475*, 2026. URL https://arxiv.org/abs/2606.06475.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024. URL https://arxiv.org/abs/2403.07974.

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

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, et al. VinePPO: Refining credit assignment in RL training of LLMs. In *International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2410.01679.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, et al. Kimi K3: Open frontier intelligence, 2026. URL https://arxiv.org/abs/2607.24653.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35, 2022. URL https://arxiv.org/abs/2206.14858.

Chenxi Li, Wenxuan Zeng, Yun Luo, Fangchen Yu, Peng Ye, Yu Cheng, and Jun Zhang. SciWalker: Synthesizing scientific coding problems with operator graphs and execution feedback, 2026a. URL https://arxiv.org/abs/2609.30054.

Yafu Li, Runzhe Zhan, Haoran Zhang, Shunkai Zhang, Yizhuo Li, Zhilin Wang, Jiacheng Chen, Futing Wang, Xuyang Hu, Yuchen Fan, Bangjie Xu, Yucheng Su, Xinmiao Han, Chenxi Li, Haodi Lei, Yufeng Zhao, Zejin Lin, Qianjia Cheng, Tong Zhu, Xiaoye Qu, Ganqu Cui, Peng Ye, Yun Luo, Zhouchen Lin, Yu Qiao, Bowen Zhou, Ning Ding, and Yu Cheng. Achieving gold-medal-level olympiad reasoning via simple and unified scaling, 2026b. URL https://arxiv.org/abs/2605.13301.

Yujiang Li, Zhenyu Hou, Yi Jing, Jie Tang, and Yuxiao Dong. CompactionRL: Reinforcement learning with context compaction for long-horizon agents. *arXiv preprint arXiv:2607.05378*, 2026c. URL https://arxiv.org/abs/2607.05378.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023. URL https://arxiv.org/abs/2305.20050.

Bill Yuchen Lin, Ronan Le Bras, Kyle Richardson, Ashish Sabharwal, Radha Poovendran, Peter Clark, and Yejin Choi. ZebraLogic: On the scaling limits of LLMs for logical reasoning. *arXiv preprint arXiv:2502.01100*, 2025. URL https://arxiv.org/abs/2502.01100.

Yun Luo, Futing Wang, Qianjia Cheng, Fangchen Yu, Haodi Lei, Jianhao Yan, Chenxi Li, Jiacheng Chen, Yufeng Zhao, Haiyuan Wan, Yuchen Zhang, Shenghe Zheng, Junchi Yao, Qingyang Zhang, Haonan He, Wenxuan Zeng, Li Sheng, Chengxing Xie, Yuxin Zuo, Yizhuo Li, Yulun Wu, Rui Huang, Dongzhan Zhou, Kai Chen, Yu Qiao, Lei Bai, Yu Cheng, Ning Ding, Bowen Zhou, Peng Ye, and Ganqu Cui. P1-VL: Bridging visual perception and scientific reasoning in physics olympiads, 2026. URL https://arxiv.org/abs/2602.09443.

Mathematical Association of America. American Mathematics Competitions, n.d. URL https://maa.org/student-programs/amc/. Accessed September 26, 2026.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. doi: 10.1038/nature14236.

Niklas Muennighoff, Qian Liu, Armel Zebaze, Qinkai Zheng, Binyuan Hui, Terry Yue Zhuo, Swayam Singh, Xiangru Tang, Leandro von Werra, and Shayne Longpre. OctoPack: Instruction tuning code large language models. *arXiv preprint arXiv:2308.07124*, 2023. URL https://arxiv.org/abs/2308.07124.

POLARIS-Project. Polaris-V2-RL-14K. Hugging Face dataset, n.d. URL https://huggingface.co/datasets/POLARIS-Project/Polaris-V2-RL-14K. Accessed September 23, 2026.

Penghui Qi, Xiangxin Zhou, and Wee Sun Lee. Best practice critic optimization. *arXiv preprint arXiv:2608.23566*, 2026. URL https://arxiv.org/abs/2608.23566.

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. *Proceedings of the First Conference on Language Modeling*, 2024. URL https://arxiv.org/abs/2311.12022.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 1889–1897, 2015a. URL https://proceedings.mlr.press/v37/schulman15.html.

John Schulman, Philipp Moritz, Sergey Levine, et al. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015b. URL https://arxiv.org/abs/1506.02438.

John Schulman, Filip Wolski, Prafulla Dhariwal, et al. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.

Zikang Shan, Han Zhong, Liwei Wang, and Li Zhao. Bringing value models back: Generative critics for value modeling in LLM reinforcement learning. *arXiv preprint arXiv:2604.10701*, 2026. URL https://arxiv.org/abs/2604.10701.

Zhihong Shao, Peiyi Wang, Qihao Zhu, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

Mirac Suzgun, Nathan Scales, Nathanael Scharli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. Challenging BIG-Bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, 2023. URL https://aclanthology.org/2023.findings-acl.824.

Minyang Tian, Luyu Gao, Shizhuo Dylan Zhang, Xinan Chen, Cunwei Fan, Xuefei Guo, Roland Haas, Pan Ji, Kittithat Krongchon, Yao Li, Shengyan Liu, Di Luo, Yutao Ma, Hao Tong, Kha Trinh, Chenyu Tian, Zihan Wang, Bohao Wu, Yanyu Xiong, Shengzhu Yin, Minhui Zhu, Kilian Lieret, Yanxin Lu, Genglin Liu, Yufeng Du, Tianhua Tao, Ofir Press, Jamie Callan, Eliu Huerta, and Hao Peng. SciCode: A research coding benchmark curated by scientists, 2024. URL https://arxiv.org/abs/2407.13168.

Hieu Tran, Zonghai Yao, and Hong Yu. Exploiting tree structure for credit assignment in RL training of LLMs. *arXiv preprint arXiv:2509.18314*, 2025. URL https://arxiv.org/abs/2509.18314.

Peiyi Wang, Lei Li, Zhihong Shao, R. X. Xu, Damai Dai, Yifei Li, Deli Chen, Y. Wu, and Zhifang Sui. Math-Shepherd: Verify and reinforce LLMs step-by-step without human annotations. *arXiv preprint arXiv:2312.08935*, 2023. URL https://arxiv.org/abs/2312.08935.

Tianyi Wang, Yixia Li, Long Li, et al. SPPO: Sequence-level PPO for long-horizon reasoning tasks. *arXiv preprint arXiv:2604.08865*, 2026. URL https://arxiv.org/abs/2604.08865.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*, 2024. URL https://arxiv.org/abs/2406.01574.

WildEval. ZebraLogic. Hugging Face dataset, n.d. URL https://huggingface.co/datasets/WildEval/ZebraLogic. Accessed September 26, 2026.

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. URL https://arxiv.org/abs/2505.09388.

Qiying Yu, Zheng Zhang, Ruofei Zhu, et al. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025. URL https://arxiv.org/abs/2503.14476.

Yufeng Yuan, Yu Yue, Ruofei Zhu, Tiantian Fan, and Lin Yan. What's behind PPO's collapse in long-CoT? Value optimization holds the secret. *arXiv preprint arXiv:2503.01491*, 2025. URL https://arxiv.org/abs/2503.01491.

Yu Yue, Yufeng Yuan, Qiying Yu, et al. VAPO: Efficient and reliable reinforcement learning for advanced reasoning tasks. *arXiv preprint arXiv:2504.05118*, 2025. URL https://arxiv.org/abs/2504.05118.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. AGIEval: A human-centric benchmark for evaluating foundation models. *Findings of the Association for Computational Linguistics: NAACL 2024*, 2024. URL https://arxiv.org/abs/2304.06364.

A Appendix

A.1 Experimental Details and Diagnostic Metrics

Table 4: Key training hyperparameters for the Qwen3-4B-Base and Qwen3-8B-Base experiments.

Setting	Value
Rollout steps	500
Rollout batch	64×8
Actor update batch	256/step; 2/rollout
Response length / temperature	8,192/1.0
Actor learning rate	1×10^{-6}
Critic learning rate	4×10^{-6}
Optimizer	Adam; $\beta = (0.9, 0.98)$; wd $= 0.1$
PPO ratio clip / KL coefficient	0.2/0
Critic-only warm-up	20 batches

Table 5: Selected training settings for the Qwen3.5-9B scientific-coding experiments.

Setting	Value
Training steps	100
Rollouts per batch	32 problems $\times$ 8 trajectories
Response cap	20,000 tokens per subproblem
GAE γ/λ	1/1
Actor learning rate	1×10^{-6}
Critic learning rate	1×10^{-5}
Policy / value clip	0.2/0.5

Generalized advantage estimation. Using the rollout critic $V_{\phi_{\text{old}}}$, GAE computes

$$\delta_t = r_t + \gamma V_{\phi_{\text{old}}}(s_{t+1}) - V_{\phi_{\text{old}}}(s_t), \qquad \widehat{A}_t = \sum_{\ell=0}^{T-t} (\gamma\lambda)^\ell \delta_{t+\ell}, \tag{7}$$

with $V_{\phi_{\text{old}}}(s_{T+1}) = 0$. The regression target is $\widehat{G}_t = \widehat{A}_t + V_{\phi_{\text{old}}}(s_t)$, as in Equation (2).

Figure and table settings. In Figure 6, the $K = 4$ point uses anchors at $0.3/0.5/0.7/0.9$. Points show mean performance, with bands indicating variation.

FrozenLake settings. Each rollout terminates when the agent reaches the goal, falls into a hole, or reaches the maximum episode length of 8192 steps. Reaching the goal yields a return of 1, while falling into a hole or reaching the episode limit yields a return of 0.

Policy-conditioned Monte Carlo estimation. For the fixed evaluation panel, we use 64 responses and evaluate each response at 20 anchors: 19 intermediate states at relative positions $0.05, 0.10, \ldots, 0.95$, plus the completed response at position 1.0. At each intermediate state, we hold the state fixed and independently sample continuations $\tau_t^{(k)} \sim \pi(\cdot \mid s_t)$ from the same actor policy. We start with 128 continuations and add batches of 64, up to a maximum of 256, stopping earlier when the half-width of the 95% Wilson confidence interval for the empirical success rate is at most 0.04. Thus, the nonterminal anchors use $K \in \{128, 192, 256\}$. We estimate the policy-conditioned state value by averaging the realized terminal rewards, $\widehat{V}^{\pi}_{\text{MC}}(s_t) = K^{-1} \sum_{k=1}^{K} G_t^{(k)}$. We use temperature 1.0, top-p 0.95, and at most 8192 generated tokens per continuation. The terminal anchor uses its observed terminal reward $R(\tau)$ directly rather than additional sampling.

Critic diagnostic metrics. For each response i, let $\Delta v_{i,t}$ denote the change in the predicted value at evaluated state t. We define the *update energy* in Figure 3(b) as

$$E_i = \frac{T_i \overline{\Delta v}_i^2}{\sum_t \Delta v_{i,t}^2}, \qquad \overline{\Delta v}_i = \frac{1}{T_i} \sum_t \Delta v_{i,t}. \tag{8}$$

Let $v_{i,t} = V_\phi(s_{i,t})$ be the critic prediction at token t of response i, T_i its number of valid tokens, and $R_i \in \{0, 1\}$ its terminal outcome. *Response AUC* is the ROC-AUC of R_i scored by the mean prediction $\bar{v}_i = T_i^{-1} \sum_t v_{i,t}$. The *within-response variation* ratio is

$$\frac{\sum_i \sum_t (v_{i,t} - \bar{v}_i)^2}{\sum_i \sum_t (v_{i,t} - \bar{v})^2}, \qquad \bar{v} = \frac{\sum_i \sum_t v_{i,t}}{\sum_i T_i}, \tag{9}$$

where lower values indicate flatter predictions within responses. Both metrics are computed per rollout batch and summarized by stage medians in Figure 4(b). For panel (c), we unit-normalize the value-head input states within each response, subtract their response mean, and stack them into a matrix H_i. If λ_j are the eigenvalues of $H_i H_i^\top$ and $p_j = \lambda_j / \sum_k \lambda_k$, its *effective rank* is $r_{\text{eff}} = \exp(-\sum_j p_j \log p_j)$; the panel plots its empirical cumulative distribution across responses. For panel (d), let $h_{i,t}$ denote the value-head input and A_i the evaluated positions. The response-level mean gradient difference under the squared-error value loss between terminal-return and MC targets, and its RMS across N responses, are

$$\Delta g_i(A_i) = \frac{2}{|A_i|} \sum_{t \in A_i} \big(\widehat{V}^\pi_{\text{MC}}(s_{i,t}) - R_i\big) h_{i,t}, \qquad D_{\text{RMS}} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \|\Delta g_i(A_i)\|_2^2}. \tag{10}$$

A.2 Theoretical Analysis of Value Flattening

We formalize two aspects of value flattening. We first decompose critic error into response-mean and within-response components and examine how the latter affects finite-batch actor updates. We then show that dense token-level supervision introduces an implicit penalty on within-response prediction variance and compare its gradient with a reference gradient based on exact policy-conditioned state values. Finally, we explain why these finite-batch effects do not conflict with the standard invariance of the expected policy gradient to a state-dependent baseline.

Decomposing critic error. Let $i \in \{1, \dots, N\}$ index responses and $t \in \{1, \dots, m_i\}$ index evaluated states. Define the policy-conditioned value $q_{i,t} = V^\pi(s_{i,t})$, the critic prediction $v_{i,t} = V_\phi(s_{i,t})$, and, for any per-state quantity $x_{i,t}$, write its response mean as $\bar{x}_i := m_i^{-1} \sum_{t=1}^{m_i} x_{i,t}$. In particular, this defines $\bar{q}_i$ and $\bar{v}_i$. The mean squared critic error, with each response weighted equally, decomposes as

$$\mathcal{E}_{\text{point}} := \frac{1}{N} \sum_{i=1}^{N} \frac{1}{m_i} \sum_{t=1}^{m_i} (v_{i,t} - q_{i,t})^2 \tag{11}$$

$$= \underbrace{\frac{1}{N} \sum_{i=1}^{N} (\bar{v}_i - \bar{q}_i)^2}_{\mathcal{E}_{\text{mean}}} + \underbrace{\frac{1}{N} \sum_{i=1}^{N} \frac{1}{m_i} \sum_{t=1}^{m_i} \big[(v_{i,t} - \bar{v}_i) - (q_{i,t} - \bar{q}_i)\big]^2}_{\mathcal{E}_{\text{within}}}. \tag{12}$$

The two terms measure error in the response mean and error in value changes within the response, respectively. If states are instead averaged over the entire batch, response i receives weight $\alpha_i = m_i / \sum_j m_j$, and the same decomposition holds. Let

$$e_{i,t} := v_{i,t} - q_{i,t} = \bar{e}_i + e^\circ_{i,t}, \qquad \frac{1}{m_i} \sum_{t=1}^{m_i} e^\circ_{i,t} = 0, \tag{13}$$

where $\bar{e}_i$ is the response-mean error and $e^\circ_{i,t}$ is the centered within-response error.

From critic error to actor updates. Under $\gamma = \lambda = 1$, all states in a response share the same realized return. Let $R_i := R(\tau_i)$ denote that terminal reward. The critic error therefore changes the estimated advantage by $-e_{i,t}$, and its within-response component changes the relative credit assigned to states by $-e^{\circ}_{i,t}$. Let $z_{i,t} = \nabla_\theta \log \pi_\theta(a_{i,t} \mid s_{i,t})|_{\theta=\theta_{\text{old}}}$ be the log-policy gradient and $\bar{z}_i = m_i^{-1} \sum_t z_{i,t}$ its response mean. At the rollout policy, define the actor gradients obtained with the sampled critic baseline and with a reference baseline using exact policy-conditioned state values:

$$\begin{aligned} g_{\text{critic}} &:= \frac{1}{N} \sum_{i=1}^{N} \frac{1}{m_i} \sum_{t=1}^{m_i} z_{i,t}\big(R_i - v_{i,t}\big), \\ g_{\text{oracle}} &:= \frac{1}{N} \sum_{i=1}^{N} \frac{1}{m_i} \sum_{t=1}^{m_i} z_{i,t}\big(R_i - q_{i,t}\big). \end{aligned} \tag{14}$$

We define the finite-batch actor-gradient error as $\Delta g := g_{\text{critic}} - g_{\text{oracle}}$. It then decomposes exactly as

$$\Delta g = \underbrace{-\frac{1}{N} \sum_{i=1}^{N} \bar{z}_i \bar{e}_i}_{\Delta g_{\text{mean}}} \underbrace{- \frac{1}{N} \sum_{i=1}^{N} \frac{1}{m_i} \sum_{t=1}^{m_i} (z_{i,t} - \bar{z}_i) e^{\circ}_{i,t}}_{\Delta g_{\text{within}}}. \tag{15}$$

The first term is caused by error in the response mean. The second is caused by error in the relative value profile within a response. By Cauchy–Schwarz,

$$\begin{aligned} \|\Delta g_{\text{mean}}\| &\le \left(\frac{1}{N} \sum_i \|\bar{z}_i\|^2 \right)^{1/2} \mathcal{E}^{1/2}_{\text{mean}}, \\ \|\Delta g_{\text{within}}\| &\le \left(\frac{1}{N} \sum_i \frac{1}{m_i} \sum_t \|z_{i,t} - \bar{z}_i\|^2 \right)^{1/2} \mathcal{E}^{1/2}_{\text{within}}. \end{aligned} \tag{16}$$

Thus, each gradient-error component is bounded by the critic RMSE multiplied by the RMS norm of the corresponding log-policy gradient. These are norm bounds; they do not by themselves assert that the error is determined by directional alignment.

Implicit Variance Penalty in Dense Critic Supervision. In the terminal-only setting, consider the critic loss under dense token-level supervision, with each response weighted equally and using the sampled target R_i. Let $J_{i,t} = \nabla_\phi V_\phi(s_{i,t})$ and $\bar{J}_i = m_i^{-1} \sum_t J_{i,t}$ denote the critic Jacobian and its response mean. The loss can then be decomposed as

$$\mathcal{L}^{\text{sample}}_{V,\text{dense}}(\phi) := \frac{1}{N} \sum_{i=1}^{N} \frac{1}{m_i} \sum_{t=1}^{m_i} \big(v_{i,t} - R_i\big)^2 = \frac{1}{N} \sum_{i=1}^{N} \left[(\bar{v}_i - R_i)^2 + \frac{1}{m_i} \sum_{t=1}^{m_i} (v_{i,t} - \bar{v}_i)^2 \right]. \tag{17}$$

Its gradient is correspondingly

$$\nabla_\phi \mathcal{L}^{\text{sample}}_{V,\text{dense}} = \frac{2}{N} \sum_{i=1}^{N} (\bar{v}_i - R_i) \bar{J}_i + \frac{2}{N} \sum_{i=1}^{N} \frac{1}{m_i} \sum_{t=1}^{m_i} (v_{i,t} - \bar{v}_i)(J_{i,t} - \bar{J}_i). \tag{18}$$

The first term fits the mean critic prediction for each response to its sampled terminal return. The second is the gradient of the within-response prediction-variance term $m_i^{-1} \sum_t (v_{i,t} - \bar{v}_i)^2$; the corresponding term in the sampled loss penalizes deviations from the response mean and exerts within-response flattening pressure during finite-batch critic updates. This pressure does not alter the pointwise population MSE optimum, $V^\pi(s)$, for a fixed policy and realizable critic. The shared return provides no direct information about value differences among states within a response, yet applying it at every state introduces this variance penalty. The same decomposition holds for token-weighted training, with response weights α_i.

To compare the PPO update in Equation (18) with the update needed to recover the policy-conditioned value profile, consider a hypothetical regression to the exact values $q_{i,t} = V^\pi(s_{i,t})$. The corresponding gradients of Equation (12) are

$$\nabla_\phi \mathcal{E}_{\text{mean}} = \frac{2}{N} \sum_{i=1}^{N} \bar{e}_i \bar{J}_i, \quad \nabla_\phi \mathcal{E}_{\text{within}} = \frac{2}{N} \sum_{i=1}^{N} \frac{1}{m_i} \sum_{t=1}^{m_i} e^{\circ}_{i,t} (J_{i,t} - \bar{J}_i) = \frac{2}{N} \sum_{i=1}^{N} \frac{1}{m_i} \sum_{t=1}^{m_i} e^{\circ}_{i,t} J_{i,t}. \tag{19}$$

Unlike the sampled-target update above, these are reference gradients for this exact-value regression. Because $m_i^{-1} \sum_t e^{\circ}_{i,t} = 0$, the response-shared Jacobian component cancels exactly. In other words, correcting within-response error requires the state-level Jacobians to differ across states.

Expected-gradient invariance and practical scope. For any action-independent baseline $b(s_t)$,

$$\mathbb{E}_{a_t \sim \pi(\cdot|s_t)} \left[\nabla_\theta \log \pi_\theta(a_t \mid s_t) b(s_t) \mid s_t \right] = b(s_t) \nabla_\theta \sum_{a_t} \pi_\theta(a_t \mid s_t) = 0. \tag{20}$$

Exact pointwise values are therefore unnecessary for preserving the expected on-policy gradient. On a finite batch, however, centered critic errors can have a nonzero batch average when multiplied by centered log-policy gradients.

The finite-batch decomposition in Equation (15) is exact for the local gradient at the rollout policy. Across multiple PPO epochs, value errors can also alter advantage signs, magnitudes, and which branch of PPO's clipped objective is active, so baseline invariance does not imply identical clipped updates.

In summary, our analysis yields three observations:

- Reusing the same terminal return at every state introduces an implicit variance penalty on critic predictions within each response.
- Within-response critic error can change finite-batch actor updates through its interaction with centered policy gradients.
- Expected policy-gradient invariance does not imply identical practical PPO updates on finite batches.

Together, these results motivate evaluating whether critic predictions track state-value changes within responses, in addition to response-level accuracy. Combined with the temporal-correlation evidence in Section 4.2, they motivate sparse critic supervision, which applies the shared terminal return to fewer, well-separated states.

A.3 Additional Comparisons of Critic Predictions and MC Values

Figure 9 provides four prompt-matched examples from the checkpoint evaluation panel underlying Figure 4(a). For these prompts, SP^{3}O has lower centered profile MSE than PPO.

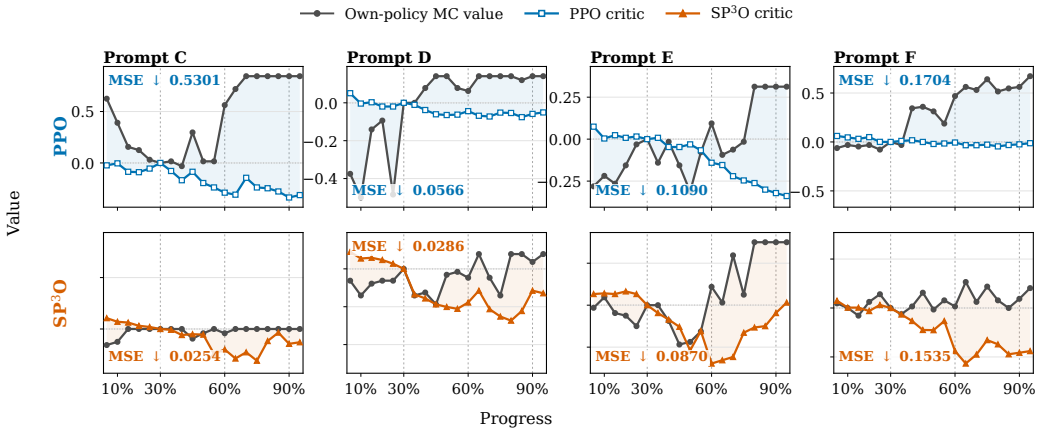

Figure 9: Additional prompt-matched value profiles. The horizontal axis is normalized response progress, and the vertical axis shows response-centered own-policy MC values and critic predictions.

A.4 SPARSE-SUPERVISION ABLATIONS

We examine how the number and placement of critic-supervision anchors affect Qwen3-4B-Base. The matched late-tail comparison is reported in Table 6. Conditional tail coverage provides a further improvement and reduces repetitive behavior.

The placement curves show that later-state coverage is generally beneficial, whereas adding anchors alone does not reliably improve performance. Figure 10 shows the corresponding online validation trajectories. Schedules with later-state coverage achieve higher validation accuracy, consistent with the placement comparison in Figure 10.

Table 6: Late-tail ablation for SP^{3}O.

Variant	Acc. (%)	Repetition (%)
SP^{3}O w/o tail anchor	44.10	18.33
SP^{3}O	**45.57**	**1.12**

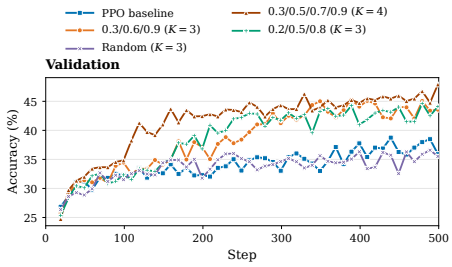

Figure 10: Performance across critic supervision anchor placements.