ADAPPO: RESOLVING DYNAMICS ASYMMETRY IN LLM ALIGNMENT VIA ADAPTIVE GATING

Anonymous authors
Paper under double-blind review

ABSTRACT

Proximal policy optimization (PPO) is widely used for aligning large language models, but it typically updates the actor and critic at every step despite their different optimization timescales. We observe and analyze a timescale asymmetry: under a local Polyak–Łojasiewicz condition, the critic converges exponentially, whereas the actor admits a sublinear stochastic convergence bound. This suggests that synchronous updates can incur redundant critic computation once the critic enters a low-gradient regime, while freezing it for too long can make its value estimates stale and degrade actor optimization. We therefore bound critic error accumulation under bounded value drift, which motivates a staleness-aware freeze schedule. Based on this analysis, AdaPPO adaptively refreshes the critic using three signals—performance, KL divergence, and entropy—evaluated jointly, with a maximum-freeze safeguard. We further introduce AdaPPO*, which augments the critic with privileged information. Experiments on four RLVR benchmarks show that AdaPPO reduces FLOPs by up to 25% and reaches target performance up to 1.7× faster in training steps. Our code is available at `https://anonymous.4open.science/r/AdaPPO-E375`.

1 INTRODUCTION

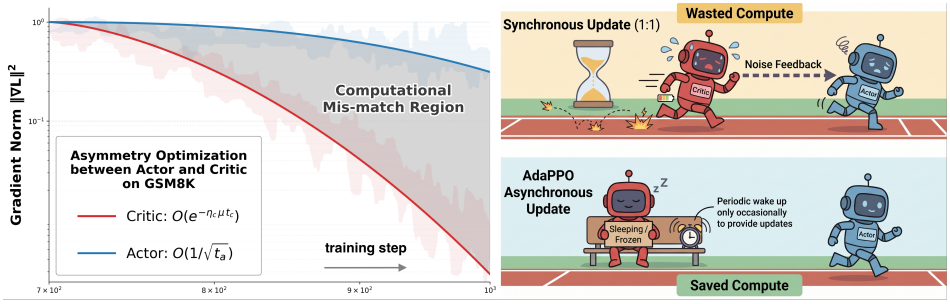


Figure 1: Empirical motivation and overview of AdaPPO. Left: Measured actor and critic gradient norms over training steps on GSM8K, showing substantially faster critic optimization. Right: Conceptual illustration of synchronous actor–critic updates (top) versus AdaPPO's adaptive freeze–refresh schedule (bottom), which avoids unnecessary critic updates while retaining value estimates for actor optimization.

Reinforcement learning is important for language model alignment and reasoning. Proximal Policy Optimization (PPO) uses a learned critic to estimate values (Schulman et al., 2017), but training this additional model is costly. A common schedule updates both actor and critic every step, although their needs differ. As Figure 1 shows, the critic may need fewer updates than the actor, making repeated fitting redundant. However, freezing the critic too long leaves its predictions outdated as the policy evolves. This raises a central question: *when should the critic be updated to balance computational cost and value accuracy?*

Recent work improves efficiency via pipeline overlap and asynchronous generation (Yan et al., 2026; Noukhovitch et al., 2025), or removes the critic via group-relative advantages (Shao et al., 2024).

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.

For critic-based methods, PPG and DCPG use phased training (Cobbe et al., 2021b; Moon et al., 2022), but predefined schedules do not adapt to the training state. The challenge remains: reduce redundant fitting while refreshing stale critics.

Our approach builds on an *optimization timescale asymmetry*: under a local PL condition, the critic converges geometrically while the actor's smooth surrogate yields a sublinear bound. This motivates unequal update frequencies. Yet the value function shifts as policy improves, so we bound critic error under bounded value drift and derive a sufficient *safe critic freeze horizon* relating freeze duration to actor optimization quality.

We introduce **AdaPPO** (Adaptive PPO), which adapts critic updates via a joint three-gate mechanism: performance (primary), KL, and entropy, plus a maximum freeze interval. The critic is frozen or refreshed based on these signals while its value predictions remain available. We further introduce **AdaPPO***, equipping the critic with privileged information.

Experiments on four RLVR benchmarks (Cobbe et al., 2021a; Liu et al., 2020; Austin et al., 2021; Guo et al., 2024) show improved accuracy–efficiency trade-offs, with ablations identifying performance feedback as the key gating signal.

Our contributions are threefold:

- **Dynamics-aware critic scheduling.** We reveal an optimization timescale asymmetry and introduce AdaPPO with a joint three-gate mechanism, plus AdaPPO* with a privileged-info critic.
- **Staleness bound on the surrogate objective.** We bound critic error accumulation under value drift and derive a sufficient staleness bound on the surrogate objective, which motivates adaptive critic scheduling without claiming a convergence guarantee for the GAE-based PPO objective.
- **Compute-efficient PPO training.** At convergence, AdaPPO achieves a $1.7\times$ speedup and uses 25% fewer FLOPs than standard PPO, while AdaPPO* provides further gains.

2 Preliminaries

2.1 PPO and Actor-Critic Setup

We consider PPO with a decoupled actor θ_A and critic θ_C over the LLM vocabulary. The two objectives are

$$J_A(\theta_A) = \mathbb{E}_t \left[\min \left(r_t \hat{A}_t,\ \text{clip}(r_t, 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right], \tag{1}$$

$$L_C(\theta_C) = \frac{1}{2}\mathbb{E}_s \left[(V_{\theta_C}(s) - V^*(s))^2 \right], \tag{2}$$

with $r_t = \pi_{\theta_A}(a_t|s_t)/\pi_{\theta_{A,\text{old}}}(a_t|s_t)$ and $\hat{A}_t$ the generalized advantage estimate. Since PPO clipping introduces nondifferentiability, our theoretical analysis uses the corresponding unclipped surrogate

$$\tilde{J}_A(\theta_A) = \mathbb{E}_t \left[r_t \hat{A}_t \right]. \tag{3}$$

When policy updates remain local, clipping is inactive; the empirical correspondence is evaluated in Appendix E.3.

2.2 Optimization Assumptions

Assumption 2.1 (L-smoothness)**.** *The smooth surrogate $\tilde{J}_A$ and the critic loss L_C are L-smooth: there exists $L > 0$ such that for all θ_1, θ_2,*

$$\|\nabla f(\theta_1) - \nabla f(\theta_2)\| \le L\|\theta_1 - \theta_2\|,$$

where f denotes either $\tilde{J}_A$ or L_C.

Assumption 2.2 (Bounded variance)**.** *The stochastic actor gradient $\hat{g}_t$ is unbiased with bounded variance:*

$$\mathbb{E}[\hat{g}_t] = \nabla_{\theta_A}\tilde{J}_A(\theta_A), \quad \mathbb{E}\left[\|\hat{g}_t - \nabla_{\theta_A}\tilde{J}_A(\theta_A)\|^2\right] \le \sigma^2,$$

where $\sigma^2 > 0$ is a constant (Appendix D.4).

We assume the critic step size satisfies $\eta_c \leq 1/L$.

2.3 Local Critic Geometry

Motivated by the overparameterized pretrained setting, we assume the critic operates in a local region satisfying the Polyak–Łojasiewicz (PL) condition: there exists $\mu > 0$ such that

$$\frac{1}{2}\|\nabla_{\theta_C} L_C(\theta_C)\|^2 \geq \mu \left(L_C(\theta_C) - L_C^*\right), \tag{4}$$

where L_C^* is the local optimum attained at $V_{\theta_C} = V^*$. This is the only geometric condition required for the critic convergence analysis; the NTK-based motivation is deferred to Appendix D.2.

3 AdaPPO: Dynamics Asymmetry and Adaptive Gating

We first establish the exponential convergence of the critic under the idealized value regression target (Sec. 3.1), then analyze the sublinear convergence of the actor in the nonconvex policy space (Sec. 3.2). The resulting optimization timescale asymmetry motivates a joint three-gate mechanism for adaptive compute allocation (Sec. 3.3), and an extension with a PI critic (Sec. 3.4).

3.1 Critic Convergence

Loss landscape and Hessian decomposition For simplicity, we assume the critic regression target is the ideal value function $V^*(s_t)$ rather than an estimated return $\hat{R}_t$. This is reasonable in a local training window, since the short-term convergence is governed by the loss curvature, not the specific target. The actual algorithm uses GAE or TD targets, but this does not affect the convergence rate.

The critic gradient is

$$\nabla_{\theta_C} L_C(\theta_C) = \mathbb{E}_s \left[(V_{\theta_C}(s) - V^*(s)) \nabla_{\theta_C} V_{\theta_C}(s) \right]. \tag{5}$$

The Hessian decomposes as

$$H_C = \nabla_{\theta_C}^2 L_C(\theta_C) = \mathbb{E}_s \left[\underbrace{\nabla_{\theta_C} V_{\theta_C}(s) \nabla_{\theta_C} V_{\theta_C}(s)^T}_{\text{Gauss-Newton term}} + \underbrace{(V_{\theta_C}(s) - V^*(s)) \nabla_{\theta_C}^2 V_{\theta_C}(s)}_{\text{residual term}} \right]. \tag{6}$$

Under the pretrained backbone, the residual term decays rapidly, so the Hessian is dominated by the Gauss-Newton term, a positive semidefinite outer-product matrix that makes the local critic loss approximately convex.

Exponential convergence under PL

Lemma 3.1 (Critic gradient norm decay)**.** *Under the PL condition and L-smoothness, gradient descent with step size $\eta_c \leq 1/L$ satisfies*

$$\|\nabla_{\theta_C} L_C(\theta_{C,t})\|^2 \leq \mathcal{O}\left(e^{-\eta_c \mu t}\right). \tag{7}$$

Proof Sketch. By the descent lemma and PL condition,

$$L_C(\theta_{C,t+1}) - L_C^* \leq (1 - \eta_c \mu) \left(L_C(\theta_{C,t}) - L_C^*\right).$$

Iterating and using $1 - x \leq e^{-x}$ gives

$$L_C(\theta_{C,t}) - L_C^* \leq (L_C(\theta_{C,0}) - L_C^*) e^{-\eta_c \mu t}.$$

Applying the L-smooth gradient upper bound $\|\nabla L_C\|^2 \leq 2L(L_C - L_C^*)$ completes the proof. □

Corollary 3.1 (Critic representational stagnation)**.** *Under the pretrained backbone and overparameterization, the critic gradient norm decays exponentially fast, so the critic enters a representational stagnation regime quickly.*

3.2 ACTOR CONVERGENCE

The actor analysis below is based on the smooth surrogate $\tilde{J}_A$, which satisfies the L-smoothness assumption. For notational simplicity, we continue to write J_A for this surrogate.

Stochastic gradient upper bound The actor seeks a first-order stationary point $\nabla_{\theta_A} J_A(\theta_A) = 0$. Under L-smoothness and bounded variance, standard nonconvex stochastic optimization gives

$$\frac{1}{t_a}\sum_{t=1}^{t_a}\mathbb{E}\|\nabla_{\theta_A} J_A(\theta_{A,t})\|^2 \le \frac{2(J^* - J_0)}{\eta_a t_a} + L\eta_a\sigma^2 \tag{8}$$

where J^* is the theoretical maximum return and $J_0 = J_A(\theta_{A,1})$. Balancing the two terms gives the optimal learning rate magnitude $\eta_a^*(t_a) \sim \sqrt{1/(\sigma^2 t_a)}$. Substituting the variance bound from Appendix D.4 yields the following sublinear bound.

Lemma 3.2 (Actor sublinear convergence bound)**.** *Under L-smoothness and bounded score-function gradients $\|\nabla_{\theta_A}\log\pi_{\theta_A}(a|s)\| \le G$, the actor's average squared gradient norm satisfies*

$$\frac{1}{t_a}\sum_{t=1}^{t_a}\mathbb{E}\|\nabla_{\theta_A} J_A(\theta_{A,t})\|^2 \le \mathcal{O}\left(\frac{1}{\sqrt{t_a}}\right), \tag{9}$$

under the optimal learning rate schedule.

3.3 ADAPTIVE GATING FROM DYNAMICS ASYMMETRY

Corollary 3.2 (Optimization timescale asymmetry)**.** *Under the above assumptions, the critic reaches a prescribed accuracy exponentially fast, whereas the actor admits only a sublinear stochastic convergence guarantee. This mismatch motivates adaptive critic scheduling.*

Proposition 3.1 (Diminishing Critic utility and overfitting risk)**.** *Let $\epsilon_C = \mathbb{E}[(V_{\theta_C}(s) - V^*(s))^2]$ be the critic value-estimation MSE. From the variance bound in Appendix D.5,*

$$\mathrm{Var}(\hat{g}_A) \le C_0 + C_1\epsilon_C.$$

Hence, the marginal benefit of reducing ϵ_C decreases as ϵ_C becomes small. This bound concerns the variance component of the surrogate objective; GAE with $\lambda < 1$ also introduces a bias term that is not controlled here. Under finite samples, excessive critic updates may additionally increase ϵ_C through overfitting.

Proof Sketch. From Appendix D.5,

$$\mathrm{Var}(\hat{g}_A) \le C_0 + C_1\epsilon_C.$$

Reducing ϵ_C by $\Delta\epsilon_C$ decreases the variance bound by at most $C_1\Delta\epsilon_C$. When ϵ_C is small, this improvement is below a prespecified tolerance. Overfitting further implies that excessive updates may increase ϵ_C, worsening gradient quality. See Appendix D.5 for details. □

Theorem 3.1 (Staleness bound on the surrogate objective)**.** *We analyze the surrogate objective defined by the stale advantages $\hat{A}_t^{stale}$, i.e., the PPO objective evaluated with a frozen critic. The bound below controls the staleness of this surrogate, not the convergence of the GAE-based PPO objective. Let ν be a fixed reference state distribution, ρ the per-step policy value drift bound (approximated by KL or value-target proxies), and $\Delta_J = J^* - J_0$. Assume $\|V^{\pi_{i+1}} - V^{\pi_i}\|_{L_2(\nu)} \le \rho$ in the freeze interval and $\sqrt{\epsilon_0} < \delta_{\text{safe}}$. Then a freeze window of length*

$$\tau \le (\delta_{\text{safe}} - \sqrt{\epsilon_0})/\rho$$

keeps the critic error below δ_{safe}^2, and the induced bound on the surrogate objective becomes

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla_{\theta_A} J_A(\theta_{A,t})\|^2 \le \frac{2\Delta_J}{\eta_a T} + L\eta_a\left[C_0 + C_1\delta_{\text{safe}}^2\right].$$

Here $\tau_{\max} = (\delta_{\text{safe}} - \sqrt{\epsilon_0})/\rho$ acts as a scheduling quantity: a longer freeze increases critic staleness, which raises policy-gradient variance and worsens the surrogate objective.

Proof Sketch. By the triangle inequality and the drift bound,

$$\|V^{\pi_{t+\tau}} - V^{\pi_t}\|_{L_2(\nu)} \leq \tau\rho.$$

Combining with the initial error gives $\epsilon_C(t+\tau) \leq (\sqrt{\epsilon_0} + \tau\rho)^2$, and substituting the variance bound into Lemma 3.2 yields the induced surrogate bound. A complete derivation is provided in Appendix D.6. □

We emphasize that Theorem 3.1 controls the staleness of the surrogate objective; it does not directly imply a convergence guarantee for the GAE-based PPO objective, which additionally involves a bias term that depends on the evolving policy distribution. The bound therefore serves as motivation for adaptive critic scheduling rather than as a formal convergence result.

We call a freeze interval **safe** if the critic value-estimation error remains below a prescribed tolerance δ^2_{safe}. Theorem 3.1 provides a sufficient staleness bound: if $\tau \leq \tau_{\max}$, then the surrogate objective's error remains bounded. When $\tau > \tau_{\max}$, the theorem no longer guarantees the bound; it does not assert that the critic has definitely become stale.

Operational Gate Criteria. We use three observable signals to approximate the latent safe-freeze boundary. These criteria are practical proxies and do not constitute formal safety guarantees; $T_{\max}$ guarantees eventual refresh, not the error tolerance of Theorem 3.1.

- **Policy Performance Gate (primary)**: triggers when $\Delta\text{Perf}_A < \tau_{\text{perf}}$, where ΔPerf_A denotes the change in policy performance (e.g., validation accuracy or RLVR reward). This approximates whether critic marginal utility is still sufficient.
- **Policy Stability Gate (auxiliary)**: triggers when $\bar{D}_{\text{KL}} > \tau_{\text{kl}}$, where $\bar{D}_{\text{KL}}$ is the expected KL divergence between the new and old policies. By the TRPO performance bound, a small $\bar{D}_{\text{KL}}$ implies a bounded expected return difference, but it does not bound the pointwise value drift ρ. Thus, KL is an operational proxy for policy drift, not a formal bound on ρ.
- **Policy Diversity Gate (auxiliary)**: triggers when $H(\pi) < H_{\min}$, where $H(\pi)$ is the policy entropy. This approximates exploration-degeneration risk, since low entropy may indicate that the critic's local advantage estimates are overfit to spurious high rewards.

Joint trigger logic. During the freeze period, the system monitors all signals. If all conditions are satisfied simultaneously, the critic is awakened; otherwise, it remains frozen. The AND logic reduces false positives, while $T_{\max}$ guarantees eventual refresh. Details are provided in Appendix E.2.

Remark 3.1 (Practical motivation for AND logic)**.** *The AND rule is the default scheduling policy: it reduces false positives and avoids frequent ineffective updates, at the cost of more false negatives. Appendix E.2 shows that both attain comparable accuracy, but the AND rule does so with fewer critic updates. Thus AND governs normal operation while $T_{\max}$ provides the safety net.*

3.4 AdaPPO* with Privileged-Information Critic

We extend the joint three-gate framework to a privileged-information setting, yielding **AdaPPO*** with a **Privileged-Info Critic (PI-Critic)**. During training, token-level logits from a teacher model or ground-truth environment features are injected into the critic as additional supervision.

Recent works Dai et al. (2023); Dherin et al. (2025); Mishra et al. (2026); Zhao et al. (2026) suggest that extra information can induce implicit gradient-descent-like adaptation without training. Injecting privileged information C^* into the critic can be viewed as providing high-quality context for value learning. By Eq. equation 5, this makes the residual $V_{\theta_C}(s) - V^*(s)$ of the PI-Critic approximately that of the ideal V^*, so that its gradient is effectively close to the ideal target gradient.

Assumption 3.1 (Privileged information improves local curvature)**.** *Let μ denote the PL constant of the standard critic and μ^* that of the PI-Critic. We assume $\mu^* > \mu$. The intuition is that privileged information provides richer supervision, reducing uncertainty in value regression and making the local loss landscape more convex—an effect consistent with operating near the ideal value target. Empirical support for this assumption is provided in Appendix E.1, where we estimate the PL constants of the standard critic and PI-Critic on GSM8K and MBPP (Figure 9); the PI-Critic consistently exhibits a higher estimated PL constant.*

Under this assumption, the PI-Critic's gradient norm decays even faster with $\mu^* > \mu$:

$$\|\nabla_{\theta_C} L_C^{\mathrm{PI}}(\theta_{C,t})\|^2 \leq \mathcal{O}\left(e^{-\eta_c \mu^* t}\right).$$

4 EXPERIMENT

4.1 EXPERIMENT SETUP

All experiments are conducted on a single server with eight NVIDIA A100-80G GPUs using OpenRLHF (Hu et al., 2024) with bf16, DeepSpeed ZeRO-3 (Rajbhandari et al., 2020), and vLLM-based (Kwon et al., 2023) rollout. The critic attaches a randomly initialized linear value head, and rewards are computed by deterministic task-specific rules during rollout, so no reward model is trained. Remaining hyperparameters are given in Appendix B.

Models & Datasets. During training, we sample multiple responses per prompt with temperature 1.0. The primary backbone is Qwen3-8B (Yang et al., 2025), supplemented by Llama-3.1-8B (Grattafiori et al., 2024) for cross-architecture comparison and Qwen3-1.7B/14B (Yang et al., 2025) for scale analysis. We evaluate on the following four RLVR benchmarks spanning diverse reasoning paradigms:

- *Math reasoning:* GSM8K (Cobbe et al., 2021a), grade-school math word problems with exact-match numeric rewards.
- *Logical reasoning:* LogiQA (Liu et al., 2020), Chinese multiple-choice logic with option-matching rewards.
- *Code generation:* MBPP (Austin et al., 2021), Python code with unit-test pass-rate rewards, randomly split into 700 training and 274 held-out problems.
- *Domain knowledge:* OWL (Guo et al., 2024), IT-operations multiple-choice QA with option-matching rewards.

Baselines. We compare AdaPPO and AdaPPO* against two online critic-update scheduling baselines and one offline reference. Standard PPO (Schulman et al., 2017) updates the critic at every step. DCPG (Moon et al., 2022) throttles updates at a fixed rate, representing a static, state-agnostic schedule. DAPO (Liu et al., 2026) is an offline two-stage method that first fits the critic to Monte-Carlo correctness probabilities and then trains the policy on the frozen critic's dataset. All methods share the same base model, data, reward, and optimization setting, and differ only in the critic-update schedule. Implementation details are provided in Appendix B.

Metrics. We evaluate quality and efficiency. Quality is measured by task reward, accuracy, and their progression over training. Efficiency is quantified by cumulative training FLOPs (actor and critic), estimated from model size and token count via the standard $C \approx 6ND$ per-token approximation (Kaplan et al., 2020), where N is the number of non-embedding parameters and D the number of tokens processed per step. The actor is counted at every step ($6ND$), while the critic is counted at every step for its forward pass ($2ND$) and additionally for a backward pass on update steps only ($4ND$), so a frozen step still incurs a forward pass. We also track KL divergence and policy entropy to monitor training stability. Accuracy is plotted in the main text, and the corresponding reward curves are reported in Appendix E.6.

4.2 MAIN RESULTS

AdaPPO achieves superior compute–performance trade-offs. As shown in Figure 2, AdaPPO achieves superior accuracy–compute trade-offs across all benchmarks. On GSM8K, it matches standard PPO at the 90% accuracy level with 25% fewer total actor-critic FLOPs, and reaches a marginally higher final accuracy (92.0% vs. 91.9%, averaged over the final 50 steps). The critic is updated in only about 54% of steps, staying frozen unless all three gates jointly fire or a safety-net fallback triggers. By contrast, DCPG(simplified) updates the critic at a high fixed rate of 70% but plateaus lower at 90.0%.

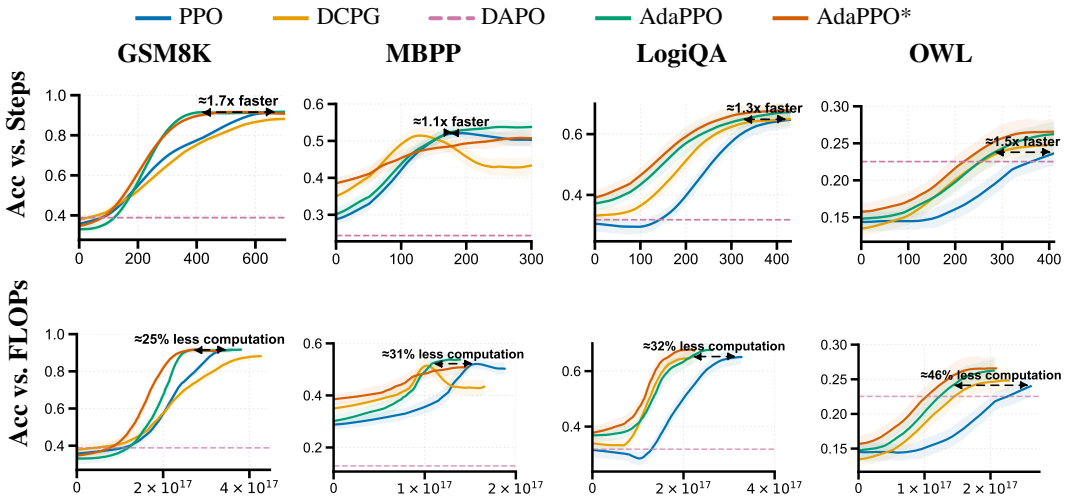


Figure 2: Compute-Performance Trade-offs of AdaPPO/AdaPPO* across reasoning benchmarks.

AdaPPO reduces redundant critic updates. Figure 3 reports the fraction of steps and of FLOPs spent updating the critic. Standard PPO updates every step, while AdaPPO updates in only about 54% of steps on GSM8K, and AdaPPO* even fewer. The FLOPs panel accumulates critic cost up to the first step where each method reaches the target accuracy (0.75 on GSM8K), so methods are compared at matched accuracy. The baseline (0.35) is the average initial accuracy of the four methods, shown only as a reference. Appendix E.4 shows comparable critic loss floors across methods, so the savings come from removing redundant updates rather than from a worse-fitted value function.

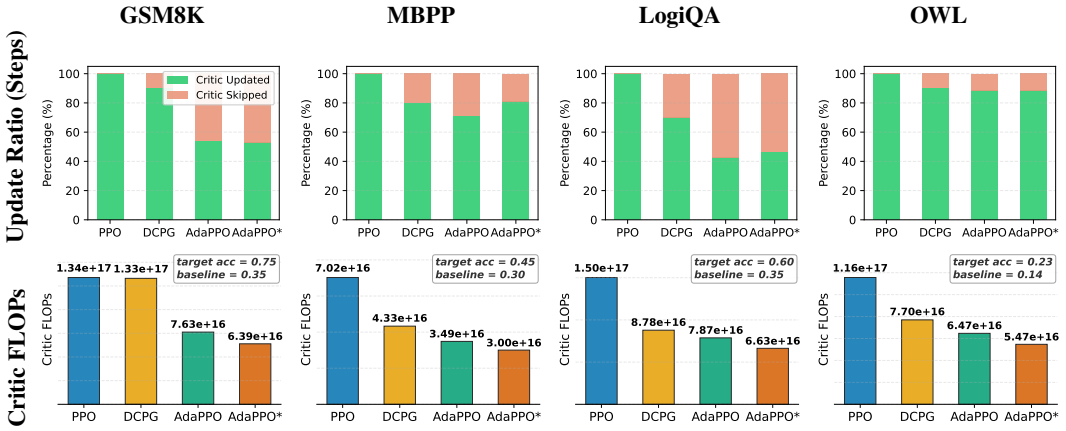


Figure 3: Critic update ratio and cumulative critic FLOPs to reach target accuracy.

4.3 Ablation Studies

We ablate the joint three-gate mechanism and study robustness across model families, initializations, and model scales.

Robustness Across Model Families and Initializations. Figure 4 shows that AdaPPO* retains its advantage over AdaPPO across the Qwen3-8B and LLaMA3.1-8B backbones and a Random-Init variant whose last four Qwen3-8B critic layers are re-initialized. A corrupted value landscape lowers the accuracy ceiling but does not reverse the relative gain.

Scaling Laws: Model Size vs. Efficiency. On GSM8K, across the Qwen3 family at 1.7B, 8B, and 14B, accuracy and reward ceilings rise monotonically with capacity, while the gap between AdaPPO

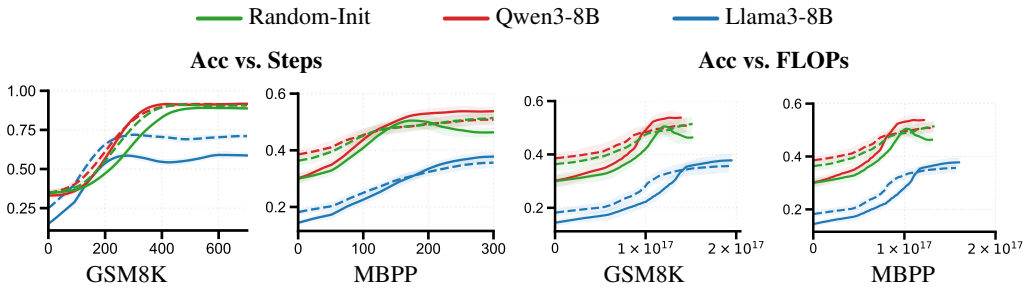


Figure 4: Robustness of AdaPPO (solid) and AdaPPO* (dashed) across model families and initializations

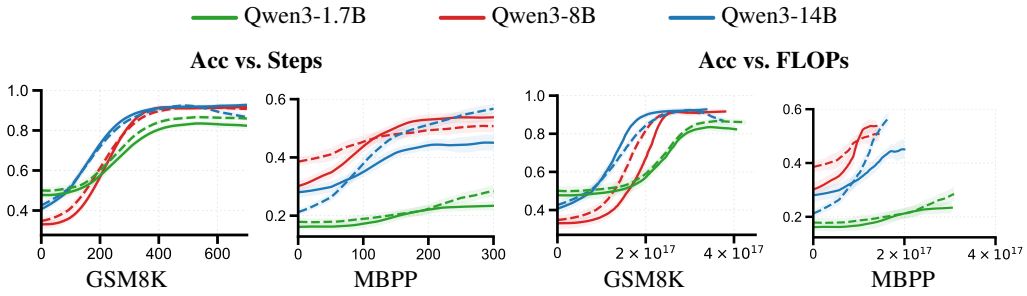


Figure 5: Scaling of AdaPPO (solid) and AdaPPO* (dashed) across model scales.

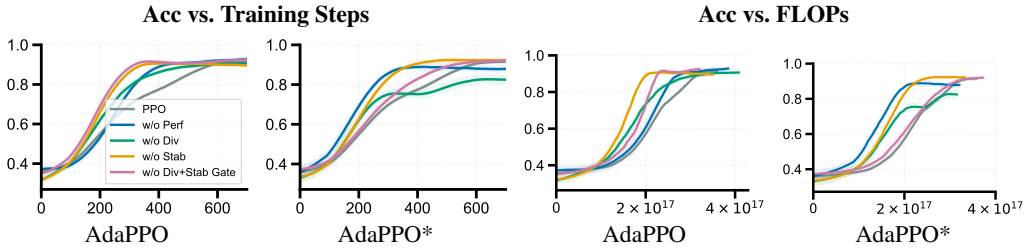


Figure 6: Ablation of the Joint Three-Gate Mechanism on GSM8K.

and AdaPPO* stays qualitatively the same (Figure 5). AdaPPO* keeps its compute advantage at every scale, with no per-scale retuning of the three gate thresholds.

Ablation of the Joint Three-Gate Mechanism. We ablate the three gates individually and in combination—w/o Performance, w/o Diversity, w/o Stability, and w/o Diversity+Stability—and compare each variant with standard PPO on GSM8K (Figure 6) and MBPP (Figure 7).

Across both datasets, the Performance Gate dominates. Removing it yields the lowest accuracy and pushes the FLOPs curves furthest right, so the same performance requires more compute. This indicates that the performance gate, rather than the auxiliary stability or diversity gates, accounts for most of the benefit of the joint AND rule.

Gate State Composition and Flow Control Analysis. Figure 8 shows cumulative gate activations on GSM8K. In the first 500 critic steps, individual gates are frequently active: KL 83.0–83.8%, performance 66.2–73.4%, entropy 54.4–57.2%. Their AND conjunction activates in only 38.4% (AdaPPO) and 36.6% (AdaPPO*) of steps, so the critic updates only when policy stagnation, drift, and entropy collapse coincide. Safety-net guards force a refresh when no gate fires within the maximum freeze window (Appendix B), bridging the gap to the overall update ratio in Figure 3. Both variants show nearly identical gate profiles, so privileged conditioning follows the same gating behavior.

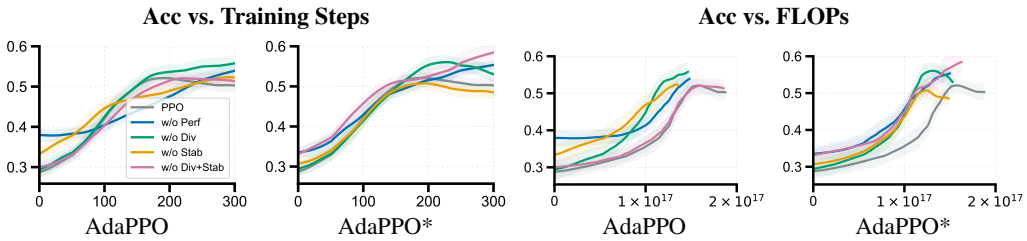


Figure 7: Ablation of the Joint Three-Gate Mechanism on MBPP.

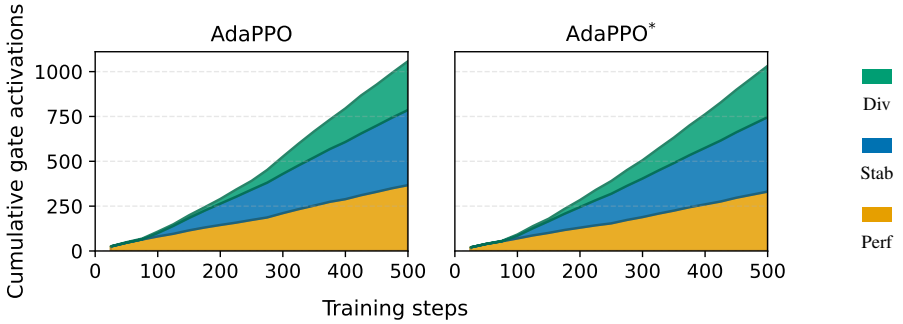


Figure 8: State composition of the Joint Three-Gate mechanism over training steps.

5 Related Work

Decoupled optimization and critic scheduling. Two-timescale stochastic approximation (Borkar, 1997) provides the foundation for analyzing coupled optimization with separated step sizes, and has been applied to actor–critic convergence, including PPO (Holzleitner et al., 2021). PPG (Cobbe et al., 2021b) separates policy and auxiliary optimization into distinct phases, while DCPG (Moon et al., 2022) updates the value function less frequently using accumulated data with value regularization; PPG Reloaded (Wang et al., 2023) identifies policy regularization and data diversity as key contributors. DAPO (Liu et al., 2026) instead freezes a critic pretrained on Monte-Carlo correctness probabilities and trains the policy offline. Unlike these predefined schedules, AdaPPO adaptively decides when to refresh the critic based on evolving training dynamics.

Efficient reinforcement learning for language models. OPPO (Yan et al., 2026) improves PPO-based RLHF throughput via pipeline overlap, and Asynchronous RLHF (Noukhovitch et al., 2025) decouples generation from learning, addressing execution efficiency and policy lag. GRPO and RLOO (Shao et al., 2024; Ahmadian et al., 2024) avoid a learned critic via group-relative and leave-one-out baselines. In contrast, AdaPPO retains the critic and adapts its update frequency to balance cost and value accuracy.

6 Conclusion

We introduced AdaPPO, which treats critic update frequency as an adaptive decision rather than a fixed schedule. By selectively refreshing the critic, AdaPPO reduces optimization overhead while preserving policy performance; AdaPPO* further improves critic learning via privileged information. Our analysis and experiments show that the critic need not be updated uniformly throughout PPO training, highlighting adaptive critic scheduling as an effective way to improve efficiency. The current gates are only indirectly linked to critic error and value drift, motivating more principled update criteria, and future work will test whether these benefits generalize beyond RLVR to broader RLHF settings.

AI USE STATEMENT

In this work, we used generative AI tools for language polishing and proofreading. We have not used generative AI tools for ideation, motivation, or the development of the core research ideas, which are entirely original to the authors, and the remaining required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for assisting with code debugging and experiment validation. We have reviewed all AI-assisted work. Specifically, all experiments were human-led with AI assistance; all data used in experiments come from public datasets and were manually verified. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Complete proofs for all theoretical results are given in Appendices A.4–A.12, and all assumptions are stated in Sections 2.2–2.3. The AdaPPO algorithm, including gate logic and safety guards, is specified in Algorithm 1 and Appendix A.2, with hyperparameters in Table 1 and Appendix A.1. All datasets are public. An anonymous version of the code is available at `https://anonymous.4open.science/r/AdaPPO-E375`, and the public repository will be released upon paper acceptance.

REFERENCES

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

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Vivek S Borkar. Stochastic approximation with two time scales. *Systems & Control Letters*, 29(5): 291–294, 1997.

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*, 2021a.

Karl W Cobbe, Jacob Hilton, Oleg Klimov, and John Schulman. Phasic policy gradient. In *International Conference on Machine Learning*, pp. 2020–2027. PMLR, 2021b.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. Why can GPT learn in-context? Language models implicitly perform gradient descent as meta-optimizers. *arXiv preprint arXiv:2212.10559*, 2023. In *Findings of ACL*, 2023.

Benoit Dherin, Michael Munn, Hanna Mazzawi, Michael Wunder, and Javier Gonzalvo. Learning without training: The implicit dynamics of in-context learning. *arXiv preprint arXiv:2507.16003*, 2025.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Hongcheng Guo, Jian Yang, Jiaheng Liu, Liqun Yang, Linzheng Chai, Jiaqi Bai, Junran Peng, Xiaorong Hu, Chao Chen, Dongfeng Zhang, Xu Shi, Tieqiao Zheng, Liangfan Zheng, Bo Zhang, Ke Xu, and Zhoujun Li. OWL: A large language model for IT operations. In *International Conference on Learning Representations*, 2024.

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

Markus Holzleitner, Lukas Gruber, José Arjona-Medina, Johannes Brandstetter, and Sepp Hochreiter. Convergence proof for actor-critic methods applied to ppo and rudder. In *Transactions on large-scale data-and knowledge-centered systems XLVIII: special issue in memory of univ. prof. dr. roland wagner*, pp. 105–130. Springer, 2021.

Jian Hu, Xibin Wu, Wei Shen, Jason Klein Liu, Zilin Zhu, Weixun Wang, Songlin Jiang, Haoran Wang, Hao Chen, Bin Chen, et al. OpenRLHF: An easy-to-use, scalable and high-performance RLHF framework. *arXiv preprint arXiv:2405.11143*, 2024.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning (ICML)*, pp. 267–274, 2002.

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

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pp. 3519–3529, 2019.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP)*, pp. 611–626, 2023.

Chaoyue Liu, Libin Zhu, and Mikhail Belkin. Loss landscapes and optimization in over-parameterized non-linear systems and neural networks. *Applied and Computational Harmonic Analysis*, 59:85–116, 2022. Also arXiv:2003.00307.

Jiacai Liu, Chaojie Wang, Chris Yuhao Liu, Liang Zeng, Rui Yan, Yiwen Sun, Yang Liu, and Yahui Zhou. Improving multi-step reasoning abilities of large language models with direct advantage-based policy optimization. *Advances in Neural Information Processing Systems*, 38, 2026.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. LogiQA: A challenge dataset for machine reading comprehension with logical reasoning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 3622–3628, 2020.

Aayush Mishra, Daniel Khashabi, and Anqi Liu. IA2: Alignment with ICL activations improves supervised fine-tuning. *arXiv preprint arXiv:2509.22621*, 2026. In *ICLR*, 2026.

Seungyong Moon, JunYeong Lee, and Hyun Oh Song. Rethinking value function learning for generalization in reinforcement learning. *Advances in Neural Information Processing Systems*, 35: 34846–34858, 2022.

Michael Noukhovitch, Shengyi Huang, Sophie Xhonneux, Arian Hosseini, Rishabh Agarwal, and Aaron Courville. Asynchronous rlhf: Faster and more efficient off-policy rl for language models. In *International Conference on Learning Representations*, volume 2025, pp. 4003–4029, 2025.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. ZeRO: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–16, 2020.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1889–1897, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/schulman15.html.

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, YK Li, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Kaixin Wang, Daquan Zhou, Jiashi Feng, and Shie Mannor. PPG reloaded: An empirical study on what matters in phasic policy gradient. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 36694–36713. PMLR, 2023. URL `https://proceedings.mlr.press/v202/wang23aw.html`.

Kaizhuo Yan, Yingjie Yu, Yifan Yu, Haizhong Zheng, and Fan Lai. Oppo: Accelerating ppo-based rlhf via pipeline overlap. In *International Conference on Learning Representations*, volume 2026, pp. 65571–65585, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026.

Table 1: Per-benchmark PPO hyperparameters and gating thresholds.

	GSM8K	**MBPP**	**LogiQA**	**OWL**
PPO hyperparameters				
Max generated tokens	1024	512	768	512
Prompts per rollout batch	16	32	16	16
Responses per prompt	8	8	4	4
Training batch size	16	32	16	16
PPO iterations	4	2	1	2
KL coefficient	0.15	0.30	0.15	0.15
Gating thresholds				
τ_{perf}	0.01	0.01	0.01	0.06
τ_{kl}	0.03	0.03	0.025	0.025
$H_{\min}$	0.25	0.10	0.60	1.10
Safety-net guards				
KL ceiling	0.32	0.35	0.30	0.30
KL surge ratio	0.25	0.25	0.25	0.25
Entropy floor	0.10	0.05	0.30	0.50

A Appendix

B Training Details

Table 1 lists the per-benchmark hyperparameters and gating thresholds; within a benchmark all methods use identical values. Unless noted in the table, all experiments use an actor learning rate of 2×10^{-6}, a critic learning rate of 9×10^{-6}, an entropy coefficient of 0.02, a clip ratio $\epsilon = 0.2$, a maximum freeze window $T_{\max} = 6$, Adam (β_1=0.9, β_2=0.95, ϵ_{Adam}=10^{-8}, zero weight decay), a cosine LR schedule (warmup 0.03, min-LR ratio 0.1), and GAE (γ=1, λ=0.95).

Baseline implementation. For DCPG, we simplify the value regularization; for DAPO, we keep the two-stage structure but omit auxiliary tricks. This isolates the scheduling effect from confounders.

The three gate signals are computed at every PPO step:

- ΔPerf_A (performance gate): the per-step accuracy is first EMA-smoothed with decay 0.5, and ΔPerf_A is the linear slope of the smoothed sequence over a sliding window of 5 steps.
- $\bar{D}_{\text{KL}}$ (stability gate): the mean KL divergence between the current and the behavior policy within the step.
- $H(\pi)$ (diversity gate): the per-step token entropy, compared raw against $H_{\min}$; an EMA of the same quantity (decay 0.9) is tracked for monitoring only.

Safety net. Independently of the AND rule, the critic is refreshed unconditionally when any guard fires: the consecutive freeze count reaches $T_{\max} = 6$, KL exceeds an upper ceiling, KL rises sharply relative to the previous step, or entropy drops below a floor. The $T_{\max}$ guard guarantees eventual consistency. Remaining guard thresholds appear in Table 1. All guards use already-introduced signals (D_{KL} and $H(\pi)$), introducing no new notation.

AdaPPO* details. AdaPPO* differs from AdaPPO in exactly one respect: the text provided to the critic. The actor, the actor's rollout distribution, the reward function, and the gating logic are unchanged, with the injection details summarized in Table 2.

Table 2: Privileged-information injection in AdaPPO*.

Item	Setting
Injected text	`[fewshot] [question] [A: answer] [response]`
Injection scope	critic only
Fewshot	2 expert demos in `Question/Response/Reward` form (one correct, one incorrect)
Answer source	ground-truth answer matched by exact prompt text
Inject rate	the privileged-information injection rate decays to a low constant floor to promote actor learning, preventing reliance on inference-unavailable information and leakage into the advantage as persistent drift

C Algorithm Details

Algorithm 1 AdaPPO: Adaptive Critic Freeze–Refresh

Require: Actor θ_A, Critic θ_C, thresholds $\tau_{\text{perf}}, \tau_{\text{kl}}, H_{\min}$, max freeze $T_{\max}$

Initialize freeze counter $c \leftarrow 0$
for iteration $t = 1, 2, \ldots$ **do**
  Collect rollout $\mathcal{D}_t$ using π_{θ_A}
  Compute advantages $\hat{A}_t$ using V_{θ_C}
  Update actor θ_A on $\mathcal{D}_t$
  $c \leftarrow c + 1$
  $g_{\text{perf}} \leftarrow \mathbb{1}[\Delta\text{Perf}_A < \tau_{\text{perf}}]$
  $g_{\text{kl}} \leftarrow \mathbb{1}[\bar{D}_{\text{KL}} > \tau_{\text{kl}}]$
  $g_{\text{ent}} \leftarrow \mathbb{1}[H < H_{\min}]$
  if $(g_{\text{perf}} \wedge g_{\text{kl}} \wedge g_{\text{ent}})$ **or** $c \geq T_{\max}$ **then**
    Update critic θ_C on $\mathcal{D}_t$
    $c \leftarrow 0$
  end if
end for

For AdaPPO, the critic additionally receives privileged information C^* as input.*

D Theoretical Proofs and Derivations

D.1 Hessian residual term approximation

The critic Hessian is $H_C = \mathbb{E}_s[\nabla_{\theta_C} V_{\theta_C}(s) \nabla_{\theta_C} V_{\theta_C}(s)^T + (V_{\theta_C}(s) - V^*(s)) \nabla^2_{\theta_C} V_{\theta_C}(s)]$. Under a pretrained backbone, the residual term decays rapidly. The main text uses the approximation that this residual is negligible near convergence, which is standard for Gauss-Newton dominance. A more rigorous treatment can impose a bounded-residual assumption; the conclusions remain unchanged.

D.2 NTK-based motivation for the PL condition

The local PL condition in Section 2.3 is motivated by the overparameterized pretrained setting. Consider the critic value network V_{θ_C} and its local gradient map $\nabla_{\theta_C} V_{\theta_C}(s)$. The Gauss-Newton component of the critic Hessian is

$$G = \mathbb{E}_s \left[\nabla_{\theta_C} V_{\theta_C}(s) \, \nabla_{\theta_C} V_{\theta_C}(s)^\top \right].$$

If G is positive definite in the local region, the critic loss is locally convex. In the neural tangent kernel (NTK) regime Jacot et al. (2018), the empirical kernel matrix remains positive definite as the network width grows, which for squared-regression problems implies that the local loss satisfies the PL condition with high probability Liu et al. (2022). We therefore treat PL as a reasonable local geometric assumption for the pretrained critic, while the formal convergence analysis only relies on PL and L-smoothness.

D.3 Fixed-learning-rate noise floor

If the actor uses a fixed learning rate $\eta_a = \eta_0$, then as $t_a \to \infty$, the bound

$$\frac{1}{t_a}\sum_{t=1}^{t_a}\mathbb{E}\|\nabla_{\theta_A} J_A(\theta_{A,t})\|^2 \le \frac{2(J^* - J_0)}{\eta_0 t_a} + L\eta_0\sigma^2$$

has a nonzero constant second term. This is the noise floor that prevents convergence to a stationary point, requiring a decaying learning rate.

D.4 Policy-gradient variance under bounded score-function gradients

The policy-gradient estimate $\hat{g}_t = \nabla_{\theta_A} \log \pi_{\theta_A}(a|s)\hat{A}_t$ satisfies

$$\mathrm{Var}(\hat{g}_t) \le \mathbb{E}_{a\sim\pi_{\theta_A}}[\|\hat{g}_t\|^2] = \sum_a \pi_{\theta_A}(a|s)\|\nabla_{\theta_A}\log\pi_{\theta_A}(a|s)\|^2\hat{A}_t^2,$$

where we used $\nabla_{\theta_A}\pi_{\theta_A}(a|s) = \pi_{\theta_A}(a|s)\nabla_{\theta_A}\log\pi_{\theta_A}(a|s)$. Assuming a bounded score function $\|\nabla_{\theta_A}\log\pi_{\theta_A}(a|s)\| \le G$, the sum simplifies to

$$\mathbb{E}_{a\sim\pi_{\theta_A}}[\|\hat{g}_t\|^2] = \mathbb{E}_{a\sim\pi_{\theta_A}}\left[\|\nabla_{\theta_A}\log\pi_{\theta_A}(a|s)\|^2\right]\hat{A}_t^2 \le G^2\hat{A}_t^2,$$

since $\sum_a \pi_{\theta_A}(a|s) = 1$. Thus $\sigma^2 \le C$ for some constant C independent of the action-space size. This bound is loose but sufficient for our purposes.

D.5 Policy Performance Gate derivation

Policy-gradient SNR.

$$\mathrm{SNR}(\hat{g}_A) \triangleq \frac{\|\nabla_{\theta_A} J_A(\theta_A)\|}{\sqrt{\mathrm{Tr}(\mathrm{Var}[\hat{g}_A])}}.$$

Advantage error propagation. Let $\delta(s) = V_{\theta_C}(s) - V^*(s)$. The advantage estimate satisfies $\hat{A}_t = A_t + \delta(s_t) + \eta_t$, where η_t is sampling noise. Ignoring higher-order terms,

$$\mathrm{Var}[\hat{g}_A] \approx \mathbb{E}[\|\nabla\log\pi\|^2]\cdot(\mathrm{Var}[\eta] + \epsilon_C).$$

For GAE with $\lambda < 1$, critic error also contributes a bias term through successive TD residuals (**?**). The variance component is analyzed above; the bias component is not controlled by our bound and remains a direction for future work.

From critic error to gradient variance bound. Assuming $\|\nabla\log\pi\| \le G$,

$$\mathrm{Var}(\hat{g}_A) \le C_0 + C_1\epsilon_C,$$

where C_0 captures sampling noise and C_1 is a constant. This inequality is the basis of Proposition 3.1.

SNR and policy performance. When SNR decreases, policy improvement stalls, reflected in $\Delta\mathrm{Perf}_A$ below threshold. This relationship is empirical and does not rely on the quantitative bound above.

D.6 Proof of Theorem 3.1

We expand the proof sketch of Theorem 3.1 in full.

Step 1: policy value drift accumulation. By the triangle inequality and the per-step drift bound,

$$\|V^{\pi_{t+\tau}} - V^{\pi_t}\|_{L_2(\nu)} \le \sum_{i=t}^{t+\tau-1}\|V^{\pi_{i+1}} - V^{\pi_i}\|_{L_2(\nu)} \le \tau\rho.$$

Step 2: critic staleness bound. Combining the initial critic error $\|V_{\theta_C} - V^{\pi_t}\|_{L_2(\nu)} \leq \sqrt{\epsilon_0}$ with Step 1,

$$\|V_{\theta_C} - V^{\pi_{t+\tau}}\|_{L_2(\nu)} \leq \sqrt{\epsilon_0} + \tau\rho.$$

Squaring gives $\epsilon_C(t+\tau) \leq (\sqrt{\epsilon_0} + \tau\rho)^2$. If $\tau \leq \tau_{\max} = (\delta_{\text{safe}} - \sqrt{\epsilon_0})/\rho$, this further implies $\epsilon_C(t+\tau) \leq \delta_{\text{safe}}^2$.

Step 3: induced actor optimization bound. Substituting $\text{Var}(\hat{g}_A) \leq C_0 + C_1\epsilon_C(t+\tau) \leq C_0 + C_1\delta_{\text{safe}}^2$ from Appendix D.5 into the actor convergence bound of Lemma 3.2 yields

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla_{\theta_A} J_A(\theta_{A,t})\|^2 \leq \frac{2\Delta_J}{\eta_a T} + L\eta_a\left[C_0 + C_1\delta_{\text{safe}}^2\right].$$

D.7 Interpretation of the actor sublinear bound

Under the optimal learning rate, the actor's average squared gradient norm decays as $\mathcal{O}(1/\sqrt{T})$. This is the interpretation of Lemma 3.2.

D.8 Policy Stability Gate derivation

Policy performance bound. By Kakade & Langford and TRPO Kakade & Langford (2002); Schulman et al. (2015),

$$|J(\pi_{\text{new}}) - J(\pi_{\text{old}})| \leq \frac{2\gamma\epsilon}{(1-\gamma)^2}\sqrt{\frac{1}{2}\bar{D}_{\text{KL}}},$$

where $\bar{D}_{\text{KL}} = \mathbb{E}_{s\sim\rho^{\pi_{\text{old}}}}[D_{\text{KL}}(\pi_{\text{new}}\|\pi_{\text{old}})]$.

Relationship to value drift. The above bound controls the expected return difference, not the pointwise value drift ρ used in Theorem 3.1. Therefore, $\bar{D}_{\text{KL}}$ serves only as an operational proxy for policy drift.

D.9 Policy Diversity Gate mechanism

DAPO Liu et al. (2026) and Skywork He et al. (2025) identify entropy collapse as a core bottleneck in LLM RLHF. Our analysis suggests critic staleness is a key driver: when the actor finds a high-reward shortcut, a stale critic overestimates its advantage, causing deterministic collapse. The diversity gate monitors entropy as an indirect detector and triggers critic recalibration.

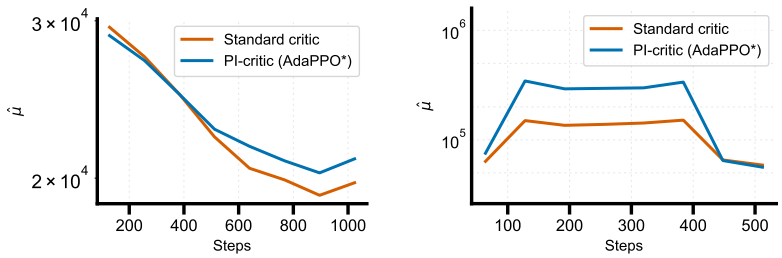
E More Experimental Details

E.1 On the $\mu^* > \mu$ assumption

The assumption $\mu^* > \mu$ is not derived from the residual inequality; it is introduced independently and validated empirically (figure 9). We estimate PL constants by $\mu_{\text{est}} = \|\nabla L(\theta)\|^2/[2(L(\theta) - L^*)]$ on a fixed set of states, and compare the standard critic and the PI-Critic on the same checkpoints. A consistently higher μ_{est} for the PI-Critic supports the assumption.

E.2 AND vs. OR trigger logic comparison

We compare the AND and OR trigger logic under their default thresholds. As shown in Figure 10, both variants obtain comparable final accuracy on GSM8K and MBPP, indicating that the AND/OR form of the trigger condition is not a performance-critical design choice. The AND gate attains this accuracy while updating the critic less frequently.



GSM8K MBPP

Figure 9: PL Constants on GSM8K and MBPP.

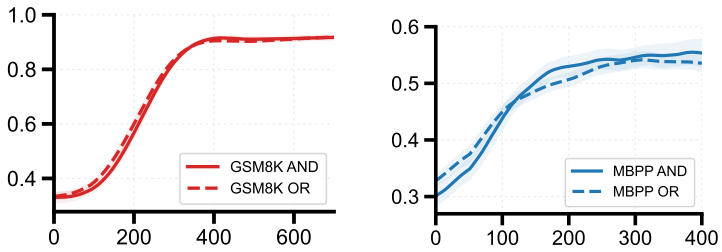


Figure 10: Comparison of AND and OR Trigger Logic on GSM8K and MBPP.

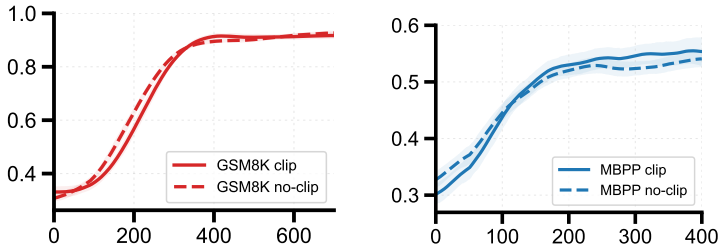


Figure 11: Ablation of the PPO Clipping Mechanism on GSM8K and MBPP.

E.3 Clip vs. no-clip ablation

We ablate the PPO clipping mechanism to verify that our analysis, derived for a smooth surrogate objective, remains informative for the clipped objective used in practice. Figure 11 compares standard PPO (clip, $\epsilon = 0.2$) with an unclipped variant on both datasets. Both variants converge to comparable final accuracy, indicating that empirical behavior is not governed by the clipping operator. This supports the qualitative transfer of our theoretical results.

E.4 Critic Loss and Cumulative Update Ratio

Figure 12 reports the critic loss together with the cumulative critic update ratio. Across all four benchmarks, every method reaches a comparable loss floor, while the cumulative update ratio required to get there differs widely: standard PPO updates the critic once per step, DCPG at a fixed fraction of steps, and AdaPPO/AdaPPO* markedly less. The critic is thus fitted equally well at very different update ratios, which underlies the FLOPs reduction reported in Figure 3.

E.5 Weight-Update Similarity of the Critic

To verify the schedule at the parameter level, we record the last Transformer MLP block every 10 steps and compute window increments

$$\delta W_t = W_t - W_{t-1},$$

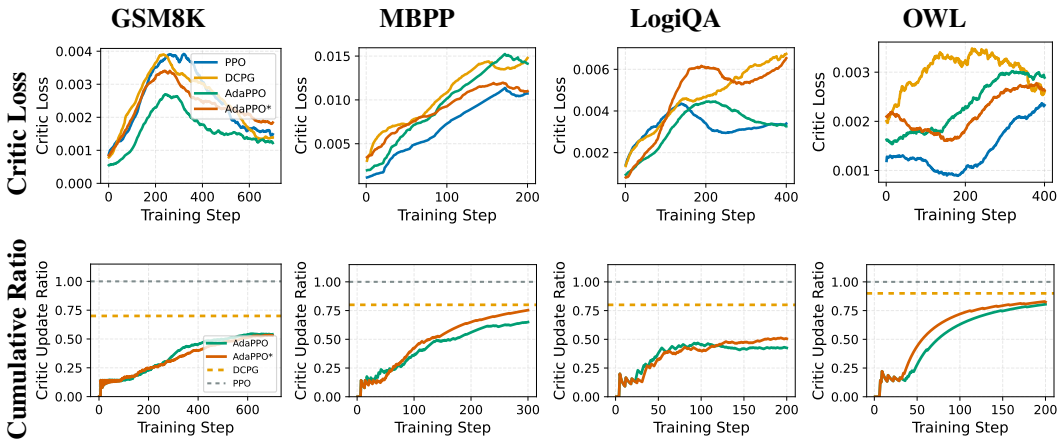


Figure 12: Critic loss and cumulative critic update ratio across all four benchmarks.

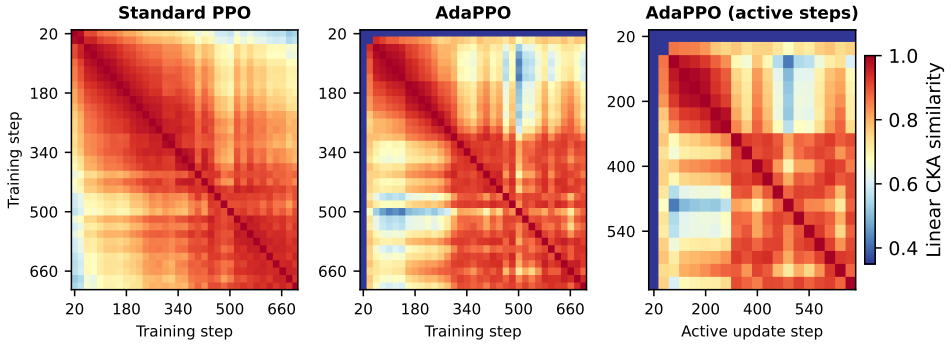


Figure 13: Linear CKA between consecutive critic weight increments on GSM8K for PPO, AdaPPO, and AdaPPO restricted to active update steps.

together with their pairwise linear CKA (Kornblith et al., 2019) (Figure 13). Standard PPO shows high off-diagonal similarity, indicating redundant computation: the critic updates every step but successive windows repeat the same direction. AdaPPO retains the diagonal block structure and develops cooler bands where gates withheld updates, consistent with saved computation. Restricting to active update steps removes these bands and leaves a dense, mutually similar block, confirming that the kept updates are non-redundant.

E.6 Reward Curves for All Benchmarks

Figure 14 reports the reward–step and reward–FLOPs curves for the main comparison. Figures 15–18 report the corresponding curves for the robustness, scaling, and gate-ablation studies. These complement the accuracy-based figures in the main text, and the conclusions hold for the reward metric as well.

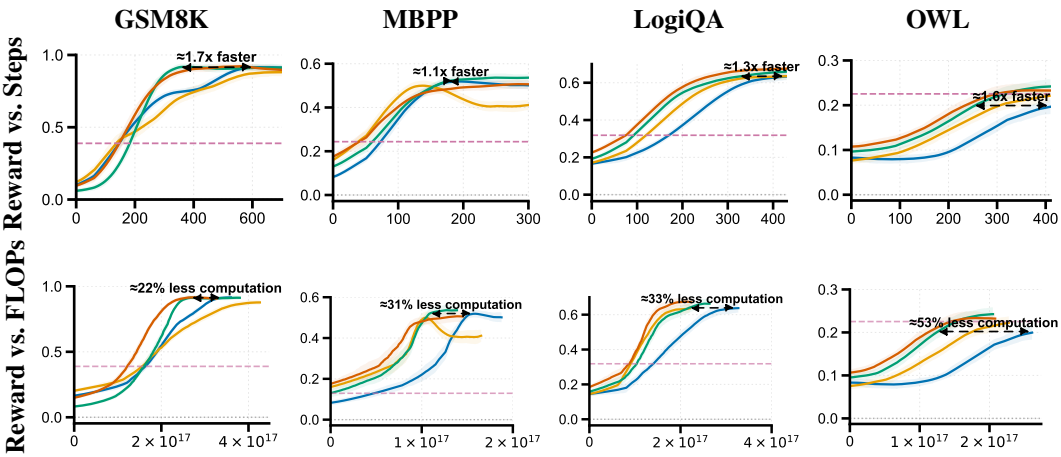


Figure 14: Reward–compute trade-offs of AdaPPO/AdaPPO* across reasoning benchmarks.

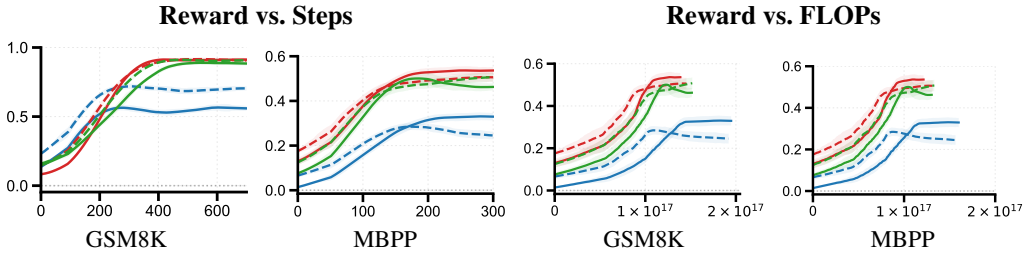


Figure 15: Reward curves for the robustness study (model family and initialization).

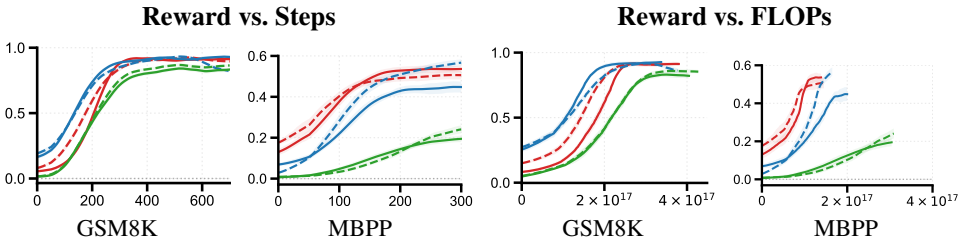


Figure 16: Reward curves for the model-scale study.

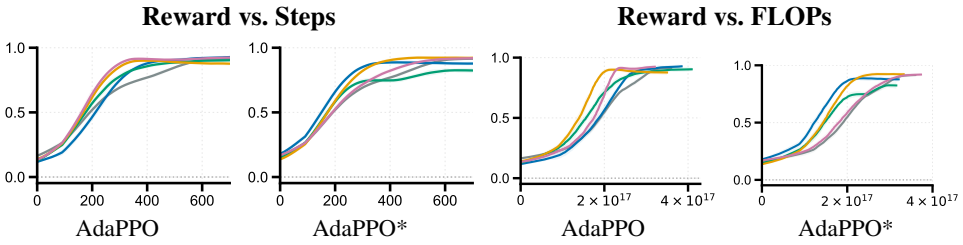


Figure 17: Reward curves for the three-gate ablation on GSM8K.

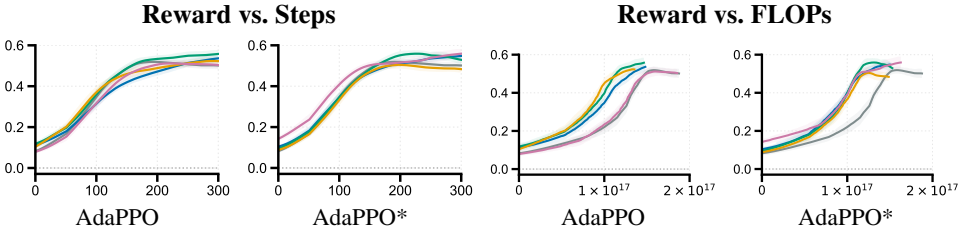


Figure 18: Reward curves for the three-gate ablation on MBPP.