ENABLING TOKEN-WISE IMPORTANCE SAMPLING IN ASYNCHRONOUS RLVR WITH PROXIMAL KL REGULARIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

In asynchronous reinforcement learning with verifiable rewards (RLVR), rollout generation overlaps with policy optimization, so training responses may come from earlier policy versions. Policy-gradient methods commonly reuse these responses with token-wise importance weighting, avoiding products of likelihood ratios across long responses. Under appropriate conditions, we prove that full future KL credit preserves the exact proximal optimizer as the token-wise update's unique stationary policy. This holds despite gradient bias, even when responses come from a mixture of older policies. The update requires neither full-response importance weights nor a fitted critic. We further establish convergence of tabular continuous-time population updates under persistent coverage and time-varying behavior policies. These results motivate *Proximal KL*, a practical asynchronous RLVR recipe that separates the proximal reference from the sampling policy and uses bounded future KL credit in place of ordinary PPO clipping. We evaluate *Proximal KL* on Qwen3-1.7B and Qwen3-4B, finding improvements over our DAPO control: higher off-policy MATH500 and MinervaMath accuracy at both tested learning rates for 1.7B and higher AIME24 and AIME25 mean@16 and pass@16 at 4B. Ablations show better preservation of AIME mean@16 as rollout lag increases and reduced MATH500 sensitivity to baseline perturbations. The horizon sweep favors a finite future window over both tested extremes, while fixed-policy measurements show higher correction variance at longer tested horizons.

1 INTRODUCTION

Policy-gradient methods for reinforcement learning with verifiable rewards (RLVR), such as GRPO and DAPO, use token-wise importance ratios to learn from verifier-scored responses (Shao et al., 2024; Yu et al., 2025). Asynchronous systems make this choice consequential: generation overlaps policy optimization, so an update may use responses sampled by earlier policy versions (Fu et al., 2025; Bai et al., 2026). Exact off-policy correction for a response-level objective multiplies importance ratios across the response. The product can have high sampling-weight variance for long responses, whereas token-wise importance weighting avoids it but generally biases the gradient. We study the conditions under which this familiar critic-free update retains the proximal target.

We study a proximal objective that regularizes expected reward by reverse KL to a reference policy held fixed during an optimization round (Geist et al., 2019; Tomar et al., 2020). The reference defines the improvement target; the behavior policy supplies responses and may be stale. Separating these roles follows Hilton et al. (2022) and supports asynchronous optimization. For autoregressive responses, the regularized policy gradient includes the current and all future token log-ratio costs, as expressed by Future-KL credit (Yao et al., 2026). We combine this full future credit with one token-wise importance ratio.

Our main result explains why token-wise importance weighting can remain effective despite off-policy gradient bias: with full future KL credit, the biased population update preserves the exact proximal optimizer as its unique stationary policy. Under coverage and local expressiveness, this holds for any supported stale behavior policy or mixture of actor versions, even though the expected

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.

update can differ from the exact proximal gradient. Soft path consistency from PCL and Trust-PCL supplies the identity at the optimizer (Nachum et al., 2017; 2018); we show that the expected sampled-reward policy-gradient update vanishes precisely at that policy. We further prove convergence of tabular population flow under persistent coverage (Appendix D).

Guided by this result, we develop *Proximal KL Regularization* (*Proximal KL*). The practical update records each token's behavior probability, periodically refreshes the proximal reference, uses a token-wise importance ratio, and subtracts bounded future KL costs from a group-relative reward advantage. A log-ratio bound and a tunable horizon bound the sampled credit, with ordinary PPO clipping removed. These choices give a practical bias–variance trade-off and are evaluated empirically (Section 4).

Contributions. To our knowledge, we provide the first stationary-policy guarantee for critic-free token-wise importance weighting with full future KL credit under supported stale behavior mixtures. Under the stated assumptions, the biased population update retains the exact proximal optimizer as its unique stationary policy; we also establish convergence of tabular population flow. This analysis motivates *Proximal KL*, an asynchronous RLVR recipe with a separate proximal reference and bounded future KL credit. In off-policy training, the method improves MATH500 and Minerva-Math greedy pass@1 at both tested learning rates for 1.7B, and AIME24 and AIME25 mean@16 and pass@16 at 4B (Table 1). The lag comparison and ablations examine sensitivity to stale sampling and baseline perturbations, the role of KL feedback in update stability, and the effect of the future-credit horizon (Section 5.2).

2 Related Work

Proximal updates. Proximal methods control policy changes relative to a reference policy, as in TRPO (Schulman et al., 2015). Regularized MDPs and mirror-descent policy optimization formalize KL-based updates (Geist et al., 2019; Tomar et al., 2020). Hilton et al. (2022) distinguish the proximal reference from the behavior policy, a separation used by AReaL and AsyncWebRL for asynchronous optimization (Fu et al., 2025; Bai et al., 2026). FRPO derives the on-policy Future-KL gradient identity for autoregressive regularization (Yao et al., 2026). Our population objective specializes this identity to linear reward utility and a fixed proximal reference (Appendix C.2). We characterize the stationary policy when full future credit is combined with a single current-token ratio under stale behavior mixtures.

PCL establishes soft path consistency and minimizes the corresponding errors on off-policy trajectories (Nachum et al., 2017). Trust-PCL extends this approach to relative-entropy regularization against an updated reference (Nachum et al., 2018). Soft Q-learning learns soft Bellman values and an associated sampling policy (Haarnoja et al., 2017); Unified PCL represents policy and value with one Q model (Nachum et al., 2017). Schulman et al. (2017a) derive regularized Monte Carlo PG and relate it to soft Q-learning with value fitting. We characterize the stationary policy of a score-function update with token-wise importance weighting, sampled reward, and full future credit, without fitting values or minimizing a consistency residual (Section 3).

Policy-gradient methods. REINFORCE uses sampled policy scores (Williams, 1992), while PPO uses a clipped policy-gradient surrogate (Schulman et al., 2017b). For language-model reasoning, GRPO constructs group-relative advantages (Shao et al., 2024), and DAPO develops clipping and sampling choices (Yu et al., 2025).

V-trace and Retrace correct bootstrapped value or Q targets for off-policy data using truncated importance traces (Espeholt et al., 2018; Munos et al., 2016). V-trace also weights its actor update by a current-action ratio. These methods establish fixed points for corrected value learning. Our result instead characterizes the unique stationary policy of a critic-free policy-gradient update using sampled rewards, full future KL credit, and current-token importance weighting.

3 Token-Wise Importance Weighting and the Proximal Target

Asynchronous RLVR learns from responses generated by earlier policy versions. Our central question is whether token-wise importance weighting can preserve the proximal optimization target when

learning from these off-policy responses. Although a single ratio per token generally biases the gradient, we show that combining it with full future KL credit preserves the exact proximal optimizer as the population update's unique stationary policy under the conditions stated below.

The proximal framework is also beneficial in the on-policy setting (Schulman et al., 2015; Tomar et al., 2020). For binary rewards, its exact unrestricted optimizer improves every prompt whose success probability lies strictly between zero and one (Proposition C.1). Appendix C.1 illustrates the benefit of proximal regularization in a shared-parameter model, and the on-policy results in Section 5.1 provide supporting evidence for its practical value. These benefits motivate the proximal regularization approach we study here.

We consider expected reward regularized by reverse KL (Geist et al., 2019; Tomar et al., 2020), with the reference policy updated between optimization rounds. Let the prompt set and generation tree be finite, with $x \sim \mathcal{D}$ and terminal reward $R(x,\tau) \in \{0,1\}$. The current policy π_θ is optimized relative to $\pi_k = \pi_{\text{prox}}$, fixed during round k; responses are sampled from π_{behav}:

$$J_k(\theta) = \mathbb{E}_{x\sim\mathcal{D},\,\tau\sim\pi_\theta}[R(x,\tau)] - \beta\mathbb{E}_{x\sim\mathcal{D}} D_{\text{KL}}(\pi_\theta(\cdot \mid x)\|\pi_k(\cdot \mid x)), \qquad \beta > 0. \tag{1}$$

Its maximizer over response policies is $\pi_k^*(\tau \mid x) = \pi_k(\tau \mid x)e^{R(x,\tau)/\beta}/Z_k(x)$, where $Z_k(x) = \mathbb{E}_{\tau\sim\pi_k(\cdot|x)}e^{R(x,\tau)/\beta}$. This well-known Gibbs characterization follows from the objective-gap identity in Equation (C.3), included for completeness.

We then ask whether this proximal improvement target can be retained under asynchronous sampling. AReaL, StreamRL, and AsyncWebRL overlap generation and policy updates to reduce idle time (Fu et al., 2025; Zhong et al., 2025; Bai et al., 2026). The reference policy defining the objective is therefore distinguished from the behavior policy defining the sampling distribution.

Differentiating the response-level objective gives the Future-KL policy-gradient expression (Schulman et al., 2017a; Yao et al., 2026): each token's score function is weighted by the reward advantage minus current and future log-ratio costs. A current-token penalty alone generally omits part of the regularization gradient (Appendix C.2). For $\tau = (a_1,\ldots,a_T)$ and $h_i = (x, a_{<i})$, the proximal importance ratio and its relation to off-policy importance weighting are

$$\rho_i = \frac{\pi_\theta(a_i \mid h_i)}{\pi_k(a_i \mid h_i)}, \qquad \frac{\pi_\theta(a_i \mid h_i)}{\pi_{\text{behav}}(a_i \mid h_i)} = \frac{\pi_k(a_i \mid h_i)}{\pi_{\text{behav}}(a_i \mid h_i)}\rho_i. \tag{2}$$

Off-policy importance weighting corrects the sampling distribution; the future KL sum specifies the regularized return. Exact response-level importance sampling uses the product of current-to-behavior ratios in Equation (2), whose variance can grow exponentially with response length (Appendix D). Starting from the policy-gradient expression for objective (1), we replace the response-wise importance weight by one ratio for each sampled token. For the current policy θ, the expected token-wise policy-gradient field is

$$\begin{aligned} \widehat{A}_i^{\text{KL}}(\theta) &= R - b(h_i) - \beta\sum_{j=i}^{T}\log\rho_j(\theta), \\ M_{\text{behav}}(\theta) &= \mathbb{E}_{\tau\sim\pi_{\text{behav}}}\sum_i \frac{\pi_\theta(a_i \mid h_i)}{\pi_{\text{behav}}(a_i \mid h_i)}\widehat{A}_i^{\text{KL}}(\theta)\nabla_\theta\log\pi_\theta(a_i \mid h_i). \end{aligned} \tag{3}$$

Here b is a detached history-only baseline, possibly zero. The advantage is treated as a fixed coefficient within each update and recomputed after each policy update. We analyze the behavior-dependent field directly through its stationary points. It uses observed rewards and policy log probabilities without a fitted value or Q function.

Theorem 1 establishes equality of the stationary policies: current-token weighting with full future KL credit preserves the proximal optimizer. Prefixes and suffixes remain behavior-distributed, so the expected gradient away from that optimizer can differ.

Theorem 1 (Token-wise importance weighting preserves the proximal target)**.** *Fix π_k and assume the finite-support, behavior-coverage, policy-expressiveness, and population-update conditions (A1)–(A4) in Appendix D.2.1. Let $M_{\text{behav}}(\theta)$ be the expected token-wise off-policy policy-gradient field in Equation (3). For any stale behavior policy, or mixture of actor versions with their logged probabilities,*

$$M_{\text{behav}}(\theta) = 0 \quad \Longleftrightarrow \quad \pi_\theta = \pi_k^*.$$

Thus the token-wise update has the same unique stationary policy as the proximal objective, although its expected update can differ from $\nabla J_k(\theta)$ at other policies.

The proof is in Appendix D.

Remark 2. At π_k^*, full future KL credit makes the advantage depend only on the current history, through the soft-consistency identity of PCL and Trust-PCL (Nachum et al., 2017; 2018). Token-wise importance weighting then makes its score expectation vanish even with stale responses; Theorem 1 establishes that this is the unique stationary policy. Appendix D.3 establishes convergence of tabular population flow under persistent coverage and gives a sufficient inner-accuracy condition for improvement with periodic reference refresh.

From theory to practice. The theorem identifies the mechanism we use: full future KL credit preserves the proximal stationary policy under token-wise weighting. Its guarantee applies to the stated population update with full, unclipped credit and the coverage and expressiveness conditions above. Our implementation retains this update structure and uses finite horizons, bounded log-ratio costs, and group normalization to control sampled credit. These choices change the analyzed update, so we assess them through reasoning performance and training stability (Sections 4-5); Table 3 specifies the differences.

4 ALGORITHM

Our off-policy implementation uses token-wise importance weighting in AReaL with bounded future KL credit. The update uses verifier rewards, policy log probabilities, and response-group statistics without fitting a value or Q function.

Update recipe.

1. **Log behavior probabilities.** For every generated token, retain the probability assigned by the actor version that sampled it. Actor policy refreshes proceed independently of proximal reference refreshes.
2. **Refresh the proximal reference.** For off-policy training, before each global-batch update, recompute $\pi_{\text{prox}} = \pi_k$ from the current policy and hold it fixed across that batch's optimizer minibatches. Responses may come from older actor versions; their logged probabilities remain the denominators. The on-policy comparison instead holds the reference fixed for ten updates, each using fresh responses.
3. **Form the token-wise importance ratio.** At each sampled (h_i, a_i), compute $\rho_i = \pi_\theta/\pi_k$ and $\pi_\theta/\pi_{\text{behav}} = (\pi_k/\pi_{\text{behav}})\rho_i$ as in Equation (2). This corrects the sampled action without multiplying ratios across the response.
4. **Bound future KL credit.** Group-normalize the verifier rewards to obtain $\widehat{A}_i$, then subtract
$$C_i^{(H)} = \beta \sum_{j=i}^{\min(T,i+H-1)} \text{clip}(\log \rho_j, -\epsilon_F, \epsilon_F). \tag{4}$$
Here H is the number of action tokens retained in the correction, including the current token, and $H = \text{full}$ retains the entire suffix. The correction is *not* normalized by window length and satisfies $|C_i^{(H)}| \le \beta\epsilon_F \min(H, T - i + 1)$.
5. **Take the policy-gradient step.** Ascend the surrogate below using detached corrected advantages, with ordinary PPO clipping off. The retained numerical safeguards are specified below and in Appendix A.1.

The horizon H selects how far future credit extends, while ϵ_F bounds each retained log-ratio cost. These choices trade approximation bias against fluctuations in sampled credit. The horizon ablation evaluates their effect on held-out performance, and Appendix A.3 defines the fixed-policy correction-variance diagnostic.

Proximal KL uses the following ascent surrogate; training minimizes its negative:
$$\mathcal{L}_{\text{PKL}}(\theta) = \mathbb{E}_{\pi_{\text{behav}}} \sum_i \frac{\pi_\theta}{\pi_{\text{behav}}} \text{sg}[\widehat{A}_i - C_i^{(H)}]. \tag{5}$$

Positive accumulated log-ratio costs reduce the coefficient multiplying the policy score, providing feedback on policy movement without an advantage-dependent clipping gate. Full, unbounded future credit supplies the regularization gradient of J_k and gives the token-wise update its stationary-policy guarantee under the conditions of Section 3. Horizon truncation then controls the extent of this feedback in the practical update.

Our **DAPO** control adapts Yu et al. (2025) with symmetric PPO clipping and the decoupled proximal and behavior policies used in asynchronous training (Hilton et al., 2022; Fu et al., 2025):

$$\mathcal{L}_{\mathrm{DAPO}}(\theta) = \mathbb{E}_{\pi_{\mathrm{behav}}} \sum_i \frac{\pi_{\mathrm{prox}}}{\pi_{\mathrm{behav}}} \min\Big(\rho_i \widehat{A}_i, \mathrm{clip}(\rho_i, 1-\epsilon, 1+\epsilon)\widehat{A}_i\Big). \tag{6}$$

Here ϵ is the PPO clipping width (0.2 in the clipped control). PPO clipping suppresses selected score-function contributions and can change the expected update. This surrogate omits the future KL gradient, and its token-wise importance weighting leaves prefixes and suffixes under the behavior distribution. Our ablations examine the bias and conditional variance effect of clipping.

Reward-only PG is a DAPO ablation that removes ordinary PPO clipping while retaining reward advantages and the asynchronous training configuration, without adding future KL regularization:

$$\mathcal{L}_{\mathrm{PG}}(\theta) = \mathbb{E}_{\pi_{\mathrm{behav}}} \sum_i \frac{\pi_\theta}{\pi_{\mathrm{behav}}} \mathrm{sg}[\widehat{A}_i]. \tag{7}$$

Removing PPO clipping eliminates its clipping bias. Future KL credit supplies the regularization term missing from a reward-gradient estimator relative to ∇J_k (Appendix A.1). Token-wise importance weighting introduces a separate off-policy approximation, and large importance weights can amplify stochastic gradient contributions. The reward-only control isolates the observed effect of adding future KL credit.

Appendices A.1 and E specify the token-averaged loss and hyperparameters, respectively.

5 Empirical Evaluation

5.1 Main Results

Protocol. We train Qwen3-1.7B-Base and Qwen3-4B-Base (Yang et al., 2025) on our 9,453-prompt training split derived from FRPO's filtered DAPO-Math data (Yao et al., 2026; Yu et al., 2025), using AReaL (Fu et al., 2025). We compare *Proximal KL* with DAPO and use Reward-only PG to test removing ordinary PPO clipping without future KL correction (Section 4). The on-policy comparison uses fresh responses with one optimizer minibatch per batch. Its *Proximal KL* variant refreshes the proximal reference every ten updates while sampling from the current policy. Appendix E gives the shared settings.

We evaluate greedy pass@1 on MATH500 (Lightman et al., 2023), OlympiadBench (He et al., 2024), and MinervaMath (Lewkowycz et al., 2022), and mean@16 and pass@16 on AIME24 and AIME25 (Mathematical Association of America, n.d.). Mean@16 is per-response correctness; pass@16 is the fraction of problems solved at least once. Table 1 reports raw scores at the 1,000-update evaluation budget; Appendix A specifies decoding and checkpoint conventions.

Results. Table 1 supports *Proximal KL* as a more effective update across model scales and learning rates. At 1.7B, its higher MATH500 and MinervaMath greedy pass@1 at both off-policy learning rates is accompanied by stronger AIME24 mean@16 and pass@16 later in training (Figure 1). The benefit extends to 4B, where both AIME24 and AIME25 improve in mean@16 and pass@16. The three-seed 1.7B results retain the off-policy gains with limited variation in greedy pass@1 and AIME mean@16: sample standard deviations are at most 2.86% and 0.73%, respectively. Figure 1 reinforces this result: *Proximal KL* sustains higher three-seed means on all three metrics across the shared evaluations in the second half of training, with consistently smaller seed-to-seed spread in AIME24 mean@16. Appendix A.4 provides a complementary question-paired analysis across all five benchmarks.

Figure 1 shows that the advantage of *Proximal KL* persists over training. Across all shared evaluations in the second half of training, it achieves higher three-seed means in MATH500 greedy pass@1

Table 1: Accuracy evaluation (%). Off-policy uses an off-policyness cap of 4 and four minibatches per training batch; on-policy uses zero lag and one minibatch. Off-policy 1.7B results at `5e-6` report mean ± sample SD over three training seeds; all other results use one seed. Evaluation budget: 1,000 training updates for all trained rows. Scores are raw. MATH500/Olympiad/Minerva use greedy pass@1; AIME uses 16 samples per problem. Bold marks the higher mean, not statistical significance. –: not applicable.

Setting	Qwen3	LR	Method	MATH500 pass@1	Olympiad pass@1	Minerva pass@1	AIME24 mean@16	AIME24 pass@16	AIME25 mean@16	AIME25 pass@16
–	1.7B	–	Base	56.10	23.15	22.43	4.27	21.67	2.92	16.67
Off-policy	1.7B	`1e-5`	DAPO	64.20	26.41	25.37	8.96	26.67	**5.83**	20.00
Off-policy	1.7B	`1e-5`	*Proximal KL*	**65.20**	**28.34**	**28.31**	**12.08**	**33.33**	5.00	20.00
Off-policy	1.7B	`5e-6`	DAPO	67.33 ± 1.15	29.57 ± 0.73	29.04 ± 0.64	7.92 ± 0.21	30.00 ± 3.33	**5.35 ± 0.73**	**24.44 ± 5.09**
Off-policy	1.7B	`5e-6`	*Proximal KL*	**68.07 ± 1.22**	**30.81 ± 1.11**	**30.76 ± 2.86**	**8.33 ± 0.21**	**32.22 ± 1.92**	5.21 ± 0.42	23.33 ± 5.77
On-policy	1.7B	`5e-6`	DAPO	66.20	33.38	**30.88**	7.29	23.33	5.21	30.00
On-policy	1.7B	`5e-6`	*Proximal KL*	**70.20**	**33.68**	29.04	**8.13**	**30.00**	5.21	30.00
–	4B	–	Base	63.75	35.61	25.74	8.59	27.50	8.33	33.33
Off-policy	4B	`5e-6`	DAPO	**81.00**	**45.85**	39.71	16.04	33.33	9.58	26.67
Off-policy	4B	`5e-6`	*Proximal KL*	79.60	44.66	**40.07**	**16.67**	**40.00**	**11.67**	**33.33**

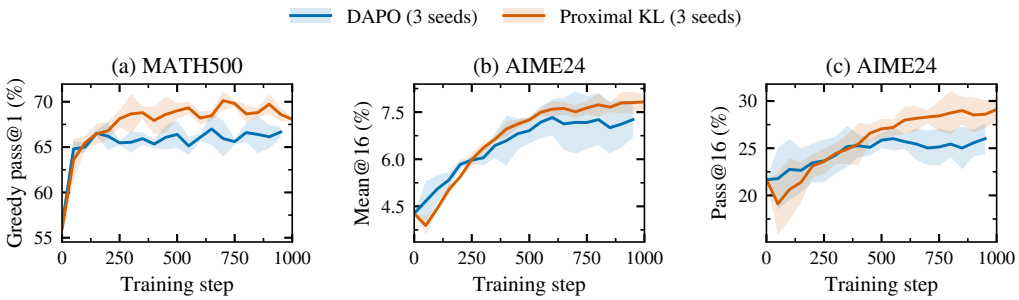

Figure 1: Qwen3-1.7B, 5×10^{-6}; full-tail *Proximal KL*, $\beta = 0.30$. Lines: three-seed means; bands: ± 1 sample SD. Evaluations align to 50-update rounds. Panel (a): raw; (b,c): per-seed EMA $\alpha = 0.20$, with a shared Base score at step 0.

and both AIME24 mean@16 and pass@16. The improvement thus appears in both per-response correctness and the fraction of problems solved within 16 attempts. On MATH500, the shaded regions showing one standard deviation across three training seeds do not overlap at most evaluations in the second half of training. For AIME24 mean@16, *Proximal KL* also has consistently smaller across-seed standard deviation throughout this interval. These trajectories show sustained gains in the three-seed comparison, beyond a single checkpoint. Appendix A.4 provides the complementary question-paired analysis across all five benchmarks.

On-policy comparison. We also evaluate fresh responses with one optimizer update per batch. Holding the proximal reference fixed for ten updates keeps KL feedback active during on-policy training. In this single-seed comparison, *Proximal KL* scores higher on MATH500, OlympiadBench, and both AIME24 metrics, lower on MinervaMath, and equally on both AIME25 metrics (Table 1). These results complement the asynchronous evaluation with a view of the recipe under fresh sampling.

5.2 Ablations

This section examines how stale sampling and the update design affect performance. The ablations show that (1) *Proximal KL* better preserves performance than DAPO as rollout lag increases; (2) it is less sensitive to baseline perturbations; (3) it is more stable than the reward-only control; and (4) the bias-variance trade-off of the tail length.

Sensitivity to rollout lag. We compare synchronous collection with in-batch reuse (lag 0) and in-batch reuse with asynchronous collection (lag 4) in Table 2. Both use four optimizer minibatches per response batch. Lag 0 therefore includes updates after the learner has moved away from the sampling policy; the on-policy comparison in Table 1 uses one update per fresh batch.
The lag comparison shows that *Proximal KL* preserves evaluation performance more consistently as responses become stale (Table 2). Increasing lag reduces DAPO's AIME24 and AIME25 mean@16, whereas both scores improve for *Proximal KL*; most of its remaining scores are also maintained or improved. These observations support the recipe's use with stale responses.

Table 2: Score changes from synchronous collection with in-batch reuse (lag 0) to asynchronous collection (lag 4): ↓ decrease; ↑ increase; → unchanged (%). Qwen3-1.7B, 5×10^{-6}. Both methods use one seed at lag 0 and a three-seed mean at lag 4.

Benchmark	DAPO	*Proximal KL*
AIME mean@16		
AIME24	↓ 1.25	↑ 1.04
AIME25	↓ 0.49	↑ 1.67
AIME pass@16		
AIME24	↑ 10.00	↑ 12.22
AIME25	↓ 2.22	↑ 6.67
Greedy pass@1		
MATH500	↑ 0.93	↓ 0.33
Olympiad	↓ 0.40	↑ 0.10
Minerva	→ 0.00	↑ 0.61

PPO clipping and baseline sensitivity. The lag comparison concerns which policy generated the responses; we next examine how the update rule treats their advantage estimates. PPO clipping can bias updates by suppressing noisy advantage contributions asymmetrically. Its active clipping side depends on the proximal ratio and advantage sign (Schulman et al., 2017b; Hilton et al., 2022): above the upper bound, positive contributions vanish. Figure 2 illustrates the resulting mean shift and reduced conditional coefficient variance. Proximal KL regularization instead subtracts accumulated policy-change costs without ordinary PPO clipping. If this additive correction is unchanged, a detached history-only baseline still cancels in the ideal unclipped, token-weighted update (Appendix A.2). This property motivates the following test of sensitivity to an empirical group baseline.

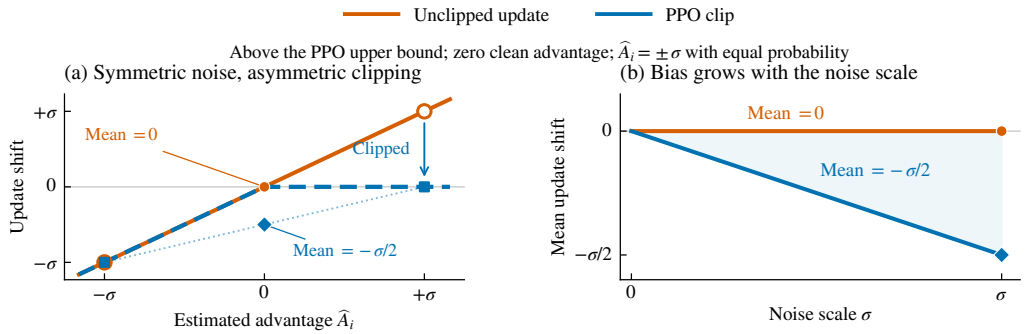

Figure 2: Above PPO's upper clipping bound, symmetric advantage noise produces a biased clipped coefficient with reduced variance; the unclipped coefficient has zero mean.

To test baseline sensitivity in training, we shift the common baseline in each group whose sampled rewards are not all equal, using normalized-reward units (Appendix A.2). The intervention preserves verifier rewards, normalization, and within-group advantage differences, with future KL correction applied as usual. It changes advantage offsets without changing which responses in a group receive higher rewards. The separation appears at 1.7B and 5×10^{-6}, *Proximal KL* loses less performance than DAPO under perturbation on all three evaluation metrics (Figure 3). These observations support the practical value of replacing the advantage-dependent clipping gate with additive future KL regularization.

Policy entropy and update stability. One might then consider removing PPO clipping without adding KL regularization. We test this alternative using Reward-only PG alongside DAPO and *Proximal KL* under a scaled setting with a 4B variant (Figure 4). The reward-only control retains the reward advantage but omits both the clipping gate and proximal KL regularization. We track policy

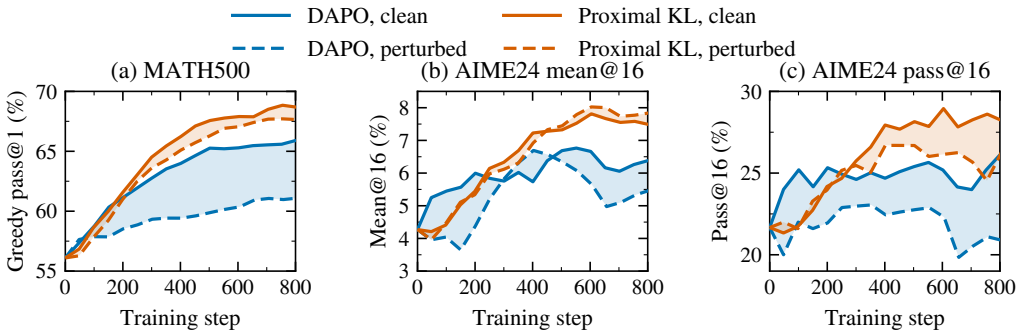

Figure 3: Baseline perturbation at 1.7B, 5×10^{-6}, lag 4. Solid/dashed: unperturbed (clean)/perturbed. Shading shows their score difference, not uncertainty. Repeated evaluations near restarts use per-metric maxima before EMA $\alpha = 0.20$.

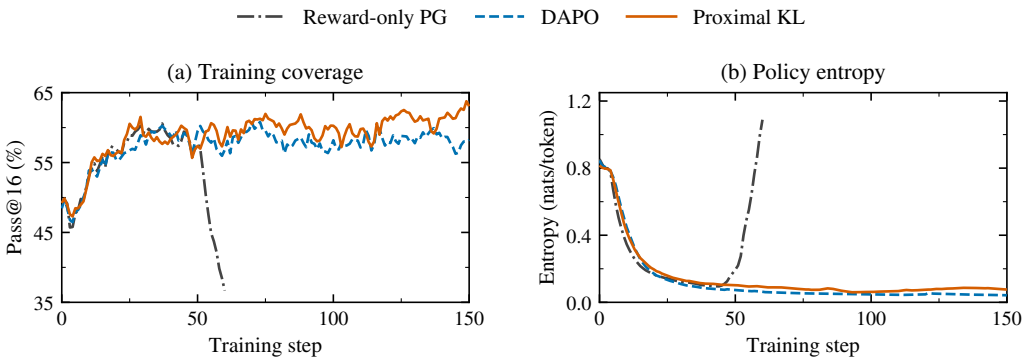

Figure 4: 4B training at 1×10^{-5}, offpolicyness 4; EMA $\alpha = 0.20$.

entropy and training pass@16, the fraction of sampled training prompts with at least one correct response in 16 attempts, to distinguish changes in entropy from changes in successful responses.

The reward-only control's pass@16 falls drastically while its entropy explodes. As a comparison, adding either the proximal KL regularization term or adding the PPO clipping maintains stable training curves. This shows that removing both terms can cause significant instability during asynchronous RL training.

Bias-variance trade-off within the future KL horizon. Selecting H balances truncation bias against correction variance. A long tail retains more future credit but can accumulate larger noise; an overly short window omits useful credit and can introduce substantial truncation bias (Figure 5). We compare four settings at 1.7B, 5×10^{-6}, and lag 4. The settings $(H, \beta) = (64, 0.075)$ and $(1, 4.80)$ match $\beta H = 4.8$, hence their clipped correction bounds. We also compare $H = 1$ and the full tail at $\beta = 0.30$.

The full-tail configuration has lower AIME24 mean@16 and pass@16 than $H = 64$ near the end of their overlapping training range, consistent with the higher correction variance observed for longer tested windows (Figure 5(b–d)). At the other extreme, using only a single token for tail summation ($H = 1$) is also troublesome: with a larger coefficient $\beta = 4.80$, its mean@16 improves slowly and remains below $H = 64$ after the first 150 updates; with $\beta = 0.30$, both mean@16 and pass@16 decline significantly later in training. These patterns are consistent with the cost of truncation bias: a strong local correction introduces a large bias during improvement, while a weak local correction has similar effect as the Reward-only PG baseline, which can cause significant instability and fail to maintain performance. The intermediate $H = 64$ window achieves a better balance in this sweep.

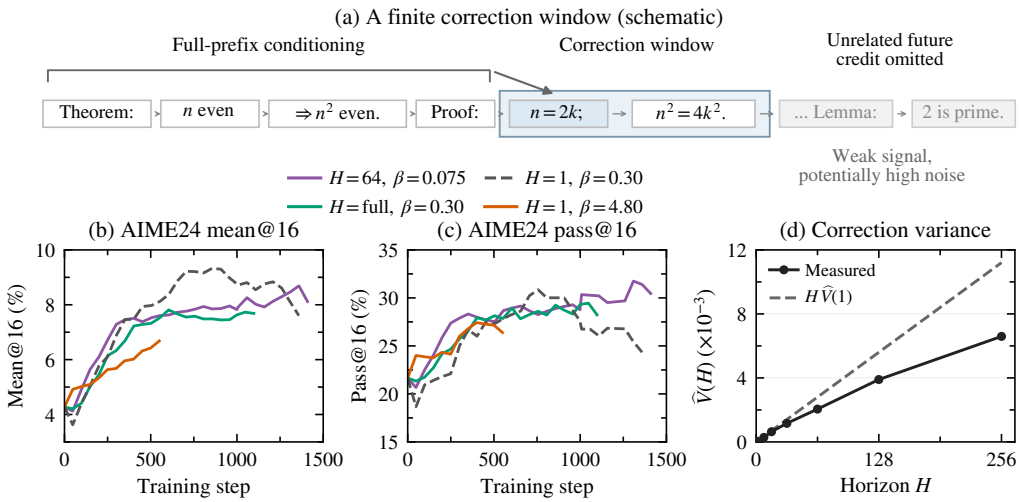

Figure 5: Future KL horizon and coefficient choices. (b,c) AIME24 mean@16 and pass@16 at 1.7B, 5×10^{-6}, lag 4; EMA $\alpha = 0.20$. Curves stop at their last observation. (d) Correction variance under fixed policies; dashed: $H\widehat{V}(1)$.

Correction variance at fixed policies. To isolate the effect of horizon length, we fix the current and reference policies and score the same responses at every tested horizon. The coefficient and log-ratio bound also remain fixed, so only the included future costs change. For each prompt, we compute the sample variance across responses of the scalar correction sum applied to the first token, then average these variances across prompts (Appendix A.3).

Figure 5(d) shows that average correction variance increases over the tested horizons, while remaining at or below the linear reference $H\widehat{V}(1)$. Even with policies and responses held fixed, longer windows therefore produce larger fluctuations in the accumulated correction. This controlled comparison provides empirical support for truncation to limit correction variability, complementing the performance comparison above.

6 Conclusion

Proximal KL uses future KL credit to make token-wise importance weighting effective for asynchronous RLVR. We prove that, under the stated coverage and expressiveness conditions, the full-tail critic-free update has the exact proximal optimizer as its unique stationary policy, and we establish convergence of tabular population flow. The practical method improves several off-policy reasoning metrics at both model scales. The lag comparison and ablations examine resilience under stale sampling, sensitivity to baseline perturbations, and update stability. The horizon comparison favors a finite future window over both tested extremes, and fixed-policy measurements show higher correction variance at longer tested horizons. Future work should choose H adaptively to retain relevant future costs while limiting fluctuations in the sampled correction.

Reproducibility Statement

Section 3 and Appendices C–D state assumptions and proofs. Sections 4–5 and Appendix A explain the algorithm and experimental design; Appendix E collects the hyperparameters.

AI Use Statement. AI tools assisted literature retrieval, manuscript drafting, mathematical arguments and checks, experiment analysis, and preparation of tables and figures.

REFERENCES

Hao Bai, Rui Yang, Chenlu Ye, Spencer Whitehead, Aviral Kumar, and Tong Zhang. AsyncWebRL: Efficient asynchronous reinforcement learning for multi-step visual web agents. *arXiv preprint arXiv:2606.05597*, 2026. URL https://arxiv.org/abs/2606.05597v3.

Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, Shane Legg, and Koray Kavukcuoglu. IMPALA: Scalable distributed deep-RL with importance weighted actor-learner architectures. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1407–1416, 2018. URL https://proceedings.mlr.press/v80/espeholt18a.html.

Wei Fu, Jiaxuan Gao, Xujie Shen, Chen Zhu, Zhiyu Mei, Chuyi He, Shusheng Xu, Guo Wei, Jun Mei, Jiashu Wang, Tongkai Yang, Binhang Yuan, and Yi Wu. AReaL: A large-scale asynchronous reinforcement learning system for language reasoning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 36256–36282, 2025. doi: 10.52202/085713-1218. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/33c00862bfa29ac72ecf630a41e19352-Paper-Conference.pdf.

Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized markov decision processes. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2160–2169, 2019. URL https://proceedings.mlr.press/v97/geist19a.html.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1352–1361, 2017. URL https://proceedings.mlr.press/v70/haarnoja17a.html.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 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/2402.14008v2.

Jacob Hilton, Karl Cobbe, and John Schulman. Batch size-invariance for policy optimization. In *Advances in Neural Information Processing Systems*, volume 35, pp. 17086–17098, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/6ceb6c2150bbf46fd75528a6cd6be793-Abstract-Conference.html.

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. *arXiv preprint arXiv:2206.14858*, 2022. URL https://arxiv.org/abs/2206.14858v2.

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/2305.20050v1.

Mathematical Association of America. MAA invitational competitions, n.d. URL https://maa.org/maa-invitational-competitions/. American Invitational Mathematics Examination (AIME) overview. Accessed September 25, 2026.

Rémi Munos, Tom Stepleton, Anna Harutyunyan, and Marc G. Bellemare. Safe and efficient off-policy reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 29, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/c3992e9a68c5ae12bd18488bc579b30d-Abstract.html.

Ofir Nachum, Mohammad Norouzi, Kelvin Xu, and Dale Schuurmans. Bridging the gap between value and policy based reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://papers.nips.cc/paper_files/paper/2017/hash/facf9f743b083008a894eee7baa16469-Abstract.html.

Ofir Nachum, Mohammad Norouzi, Kelvin Xu, and Dale Schuurmans. Trust-PCL: An off-policy trust region method for continuous control. In *International Conference on Learning Representations*, 2018. URL https://arxiv.org/abs/1707.01891v3.

John Schulman, Sergey Levine, Philipp Moritz, Michael Jordan, and Pieter Abbeel. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1889–1897, 2015. URL https://proceedings.mlr.press/v37/schulman15.html.

John Schulman, Xi Chen, and Pieter Abbeel. Equivalence between policy gradients and soft q-learning. *arXiv preprint arXiv:1704.06440*, 2017a. URL https://arxiv.org/abs/1704.06440v4.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017b. URL https://arxiv.org/abs/1707.06347v2.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300v3.

Manan Tomar, Lior Shani, Yonathan Efroni, and Mohammad Ghavamzadeh. Mirror descent policy optimization. *arXiv preprint arXiv:2005.09814*, 2020. URL https://arxiv.org/abs/2005.09814v5.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992. doi: 10.1007/BF00992696. URL https://link.springer.com/article/10.1007/BF00992696.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388v1.

Jiarui Yao, Ruida Wang, Hao Bai, and Tong Zhang. Future-KL regularized GRPO: Process-level credit assignment from f-divergence regularization. *arXiv preprint arXiv:2601.10201*, 2026. URL https://arxiv.org/abs/2601.10201v2.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems*, volume 38, pp. 113222–113244, 2025. doi: 10.52202/085713-3775. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/a4277440d50f1f15d2cb4c14f7e0c0d2-Paper-Conference.pdf.

Yinmin Zhong, Zili Zhang, Xiaoniu Song, Hanpeng Hu, Chao Jin, Bingyang Wu, Nuo Chen, Yukun Chen, Yu Zhou, Changyi Wan, Hongyu Zhou, Yimin Jiang, Yibo Zhu, and Daxin Jiang. StreamRL: Scalable, heterogeneous, and elastic RL for LLMs with disaggregated stream generation. *arXiv preprint arXiv:2504.15930*, 2025. URL https://arxiv.org/abs/2504.15930v1.

A EXPERIMENTAL DESIGN

This section specifies the implemented update and its evaluation. After defining stale-data and reporting conventions, we describe the loss, baseline perturbation, finite-horizon KL correction, and synchronous versus asynchronous rollout collection. The formal target guarantee appears separately in Appendix D.

Sampling staleness. All asynchronous experiments use a **fixed** off-policyness of 4. The on-policy main comparison uses lag 0 and one optimizer minibatch; the separate lag-0 synchronous yet off-policy control in Table 2 retains four minibatches.

Evaluation metrics. Mean@16 measures correctness per response; pass@16 measures prompts solved at least once. With independent attempts and per-prompt success probability p, expected pass@16 is $1-(1-p)^{16}$. This nonlinear transformation distinguishes repeated-attempt success from per-response correctness, so we report both metrics. MATH500, OlympiadBench, and MinervaMath use greedy pass@1; both AIME tests use 16 responses per problem. The five benchmarks contain 500, 674, 272, 30, and 30 problems, respectively. Dashes denote unavailable evaluations.

Evaluation checkpoints. Table 1 reports the 1,000-update evaluation budget for both models and methods. Scores are raw, without smoothing or performance-based checkpoint selection. Learning curves and early-training diagnostics are separate. The full-tail curve in Figure 5(b,c) continues from the 1,000-update model weights with Adam restarted.

A.1 PRACTICAL UPDATE DESIGN

Proximal and behavior policies. In off-policy training, we recompute probabilities before each global batch and freeze the proximal policy for its minibatches; the on-policy comparison instead refreshes this reference every ten updates. Stored rollout probabilities define the behavior policy. The ratio ρ_i measures movement from the proximal policy; $\pi_{\text{prox}}/\pi_{\text{behav}}$ corrects stale-action sampling. Their product is the token-wise importance weight in Equation (5).

Bias relative to the proximal objective. Fix θ, the reference policy, and the sampling law. With exact response importance correction, raw rewards, and a detached history-only baseline, a reward-gradient estimator satisfies

$$\mathbb{E}\widehat{g}_{\text{reward}} - \nabla J_k = \beta \nabla_\theta \mathbb{E}_{x\sim\mathcal{D}} D_{\text{KL}}(\pi_\theta(\cdot \mid x) \| \pi_k(\cdot \mid x)) \,. \tag{A.1}$$

The difference is the omitted KL gradient, which vanishes at the reference policy. Full future KL correction supplies it under exact sampling. Response-wide importance weighting preserves this equality off policy; token-wise importance weighting generally changes the gradient field. Group normalization and clipping the future log-ratio costs introduce further approximations.

Group normalization and KL correction. For a group of G responses, define

$$\overline{R} = \frac{1}{G}\sum_{m=1}^{G} R_m, \qquad s_R^2 = \frac{1}{G-1}\sum_{m=1}^{G}(R_m - \overline{R})^2, \qquad \widehat{A}_m = \frac{R_m - \overline{R}}{s_R + 1\times 10^{-5}}. \tag{A.2}$$

Subtract the KL correction $C_{m,i}^{(H)}$ from each token's reward advantage (Equation (4)). It includes the current token, stays within the response, and has no window-length normalization. The window selects costs; β sets their per-token strength.

Detached update coefficients. Using the group-normalized reward advantage and the future KL correction defined above, the score-function loss averaged over N_{tok} sampled action tokens is

$$\mathcal{L}^{(H)} = -\frac{1}{N_{\text{tok}}}\sum_{m,i} \text{sg}\left[\frac{\pi_\theta(a_{m,i} \mid h_{m,i})}{\pi_{\text{behav}}(a_{m,i} \mid h_{m,i})}(\widehat{A}_m - C_{m,i}^{(H)})\right] \log \pi_\theta(a_{m,i} \mid h_{m,i}). \tag{A.3}$$

Detaching this coefficient yields the score-function update in Equation (5). Normalization by sampled token count differs from the ideal response-averaged sum.

Table 3: From the stationary-policy result to the implemented update.

Ideal condition	Implemented choice	Effect on the sampled update
Full future KL tail	Horizon H, including $H =$ full	Selects how far credit extends; a finite window omits later costs.
Unclipped future log ratio	Optional log-ratio clipping	Bounds each future cost and changes its expectation when active; the choice is independent of the sampling policy.
History-only baseline	Group-normalized reward advantage	Sets the reward scale from sampled responses; the resulting field can differ from the ideal one.
Population response average	Minibatch token average	Adds sampling noise; token normalization can change the relative weights of responses.

The horizon and log-ratio bound control the size and extent of sampled future credit and can introduce approximation bias. Their effect on total update variance depends on the covariance of retained and omitted terms. Group normalization and token averaging can also change the expected field and its stationary policy.

Scope of the implementation. The implementation uses shared Transformer parameters and Adam with the estimator choices listed in Table 3. Section 3 explains their relation to the stationary-policy guarantee, and Section 5.2 evaluates clipping and future-credit horizon. Appendix E gives the settings.

A.2 Baseline Perturbation

We test sensitivity to the empirical baseline by shifting every advantage in an informative group after reward normalization:

$$\widehat{A}'_{g,m} = \widehat{A}_{g,m} - \delta I_g, \qquad \delta = -1, \qquad I_g = \mathbf{1}\{\max_m R_{g,m} > \min_m R_{g,m}\}. \tag{A.4}$$

The shift is measured in normalized advantage units, preserves within-group differences, and leaves constant-reward groups unchanged. Verifier rewards and normalization are unchanged; future KL correction is subtracted afterward, as in the clean runs.

The intervention shifts a sample-dependent group baseline; the identity below concerns a detached history-only baseline. The experiment measures sensitivity to the baseline used in training. Figure 3 compares each method with its clean reference; shading represents performance differences, not uncertainty.

Why clipping can make the baseline matter. For a frozen positive behavior correction and away from the $[0.8, 1.2]$ interval boundaries, the PPO surrogate's score coefficient is $\frac{\pi_\theta}{\pi_{\text{behav}}} q_{\text{clip}}(\rho_i, \widehat{A}_i)$, where

$$q_{\text{clip}}(\rho, A) = \begin{cases} \max(A, 0), & \rho < 0.8, \\ A, & 0.8 < \rho < 1.2, \\ \min(A, 0), & \rho > 1.2. \end{cases} \tag{A.5}$$

Above the upper boundary, centered noise taking values $\pm\sigma$ equally often leaves a coefficient with mean $-\sigma/2$ and variance $\sigma^2/4$, versus zero mean and variance σ^2 before clipping. Figure 2 illustrates this exchange of bias and fluctuation control at a fixed ratio and score. These conditional statistics neither order full gradient variances nor predict the effect of the sampled-group intervention.

For a detached history-only baseline shift $b(h)$ and positive sampling support, the corresponding unclipped change cancels:

$$\mathbb{E}_{a\sim\pi_{\text{behav}}(\cdot|h)}\left[\frac{\pi_\theta(a \mid h)}{\pi_{\text{behav}}(a \mid h)} b(h) \nabla_\theta \log \pi_\theta(a \mid h)\right] = b(h)\nabla_\theta \sum_a \pi_\theta(a \mid h) = 0. \tag{A.6}$$

An additive correction unchanged by the shift preserves this identity; the PPO clipping surrogate need not.

A.3 Paired Horizon Diagnostic

We measure how horizon length changes the scalar KL correction while holding the policies and responses fixed (Figure 5(d)). The reference and current policies are Qwen3-1.7B checkpoints at 1,000 and 1,022 updates from the same run (learning rate 1×10^{-6}, seed 1). They form a fixed diagnostic pair, not the original within-minibatch proximal pair. "Paired" means that every horizon uses the same responses; no training or new response generation is performed.

We score $P = 16$ prompts x_p, each with $M = 16$ retained responses y_{pm} of lengths T_{pm}. With $\beta = 0.30$ and $\epsilon_F = 0.05$, the first-token correction is

$$C_{pm}^{(H)} := C_1^{(H)}(x_p, y_{pm}) = \beta \sum_{j=1}^{\min(H, T_{pm})} \mathrm{clip}(\log \rho_j, -\epsilon_F, \epsilon_F), \tag{A.7}$$

where ρ_j is the current-to-reference token-probability ratio. Each response contributes one scalar sum, without normalization by length or H. All horizons use all 256 responses and stop accumulation at the response boundary. Saved text is re-tokenized identically for all horizons; nine response lengths differ from the logs by one or two tokens.

We compute the sample variance across responses separately for each prompt, then average equally across prompts:

$$\begin{aligned} \overline{C}_p^{(H)} &:= \frac{1}{M} \sum_{m=1}^{M} C_{pm}^{(H)}, \\ \widehat{V}(H) &:= \frac{1}{P} \sum_{p=1}^{P} \left[\frac{1}{M-1} \sum_{m=1}^{M} \left(C_{pm}^{(H)} - \overline{C}_p^{(H)} \right)^2 \right]. \end{aligned} \tag{A.8}$$

This measures within-prompt variation of the scalar correction, not variance across training steps or seeds, or variance of the policy gradient.

Figure 5(d) plots $\widehat{V}(H)$ in units of 1×10^{-3}, without smoothing or variance-ratio normalization. The dashed line $H\widehat{V}(1)$ is the linear reference for uncorrelated contributions with equal marginal variance before response termination. Average correction variance increases over the tested horizons while remaining at or below this reference; individual prompts need not exhibit monotonic growth.

A.4 Statistical Analysis

Table 1 reports means and sample standard deviations across three training seeds for the off-policy 1.7B comparison at 5×10^{-6}. Here we pair evaluation questions, rather than seed labels, to account for variation shared by both methods, such as question difficulty.

Evaluation protocol. We reevaluate the six models on 500 MATH500, 674 OlympiadBench, 272 MinervaMath, and 30 questions from each AIME test. The first three benchmarks use greedy decoding; AIME uses 16 responses per question, temperature 0.6, and top-$p = 0.95$. The DAPO checkpoint steps are 948, 999, and 999; all three *Proximal KL* checkpoints are at step 999. These records are separate from the aggregate evaluations in Table 1.

Paired comparison. For each question q, let x_q and y_q denote scores averaged over *Proximal KL*'s and DAPO's three models, respectively. Each model contributes correctness for greedy pass@1, the fraction of correct responses for mean@16, or an indicator of at least one correct response for pass@16. We average three pass@16 scores, without pooling attempts into pass@48. For N questions,

$$\begin{aligned} \widehat{\Delta} &= \frac{1}{N} \sum_{q=1}^{N} (x_q - y_q), \\ \widehat{\mathrm{SE}}_{\mathrm{paired}}^2 &= \frac{s_x^2 + s_y^2 - 2s_{xy}}{N}, \qquad \widehat{\mathrm{SE}}_{\mathrm{unpaired}}^2 = \frac{s_x^2 + s_y^2}{N}, \end{aligned} \tag{A.9}$$

where s_x^2, s_y^2, and s_{xy} are the sample variances and covariance across questions. Positive covariance reduces the variance of the difference. Pairing leaves its mean unchanged; differences from Table 1 arise from the separate evaluations.

We use 20,000 question-level bootstrap resamples for percentile confidence intervals. Paired resampling uses the same question indices for both methods; unpaired resampling draws them independently. Each question retains all model scores and responses. The intervals condition on the six fitted models and observed responses, rather than estimating uncertainty from new training runs. For simultaneous assessment of all seven metrics, Bonferroni-adjusted intervals use tail probability $0.05/(2 \times 7)$.

Table 4: Question-paired analysis of separate evaluations of the six models in Table 1. Differences are *Proximal KL* minus DAPO (%).

Benchmark	Metric	Difference	Unadjusted 95% CI	
			Unpaired	Paired
MATH500	greedy pass@1	+0.73	[−4.33, 5.87]	[−1.33, 2.80]
OlympiadBench	greedy pass@1	+2.67	[−1.58, 6.92]	[0.69, 4.60]
MinervaMath	greedy pass@1	+2.45	[−4.29, 9.44]	[0.00, 4.90]
AIME24	mean@16	+1.74	[−7.22, 10.76]	[−0.28, 4.24]
AIME24	pass@16	+3.33	[−16.67, 23.33]	[−2.22, 8.89]
AIME25	mean@16	+0.49	[−6.60, 7.78]	[−0.90, 2.01]
AIME25	pass@16	0.00	[−20.00, 18.89]	[−8.89, 8.89]

Results. Pairing reduces estimated question-sampling variance by 79.5%–96.1% across the seven metrics (Table 4). Only OlympiadBench's unadjusted paired 95% interval excludes zero: $[0.69\%, 4.60\%]$ for the greedy pass@1 difference. After Bonferroni adjustment across seven metrics, all intervals include zero.

B Learning-rate selection

The 1.7B DAPO sweep identifies the learning-rate candidates (Figure 6). Across recorded evaluations at labels 650–999, 5×10^{-6} has the highest average MATH500 greedy pass@1 and AIME24 pass@16; AIME24 mean@16 favors 1×10^{-5}. Table 1 therefore compares both methods at both rates. These averages use raw evaluations after the restart-adjacent consolidation described in Figure 6, without interpolation or smoothing.

C Proximal target and future KL gradient

For a fixed full-support reference π_k and $\beta > 0$, the proximal objective has an explicit response-level maximizer. We recall it and its objective-gap identity for completeness.

Proposition C.1 (Exact proximal improvement)**.** *For $\beta > 0$ and a full-support reference π_k, the unrestricted proximal maximizer is*

$$\pi_k^*(\tau \mid x) = \frac{\pi_k(\tau \mid x) e^{R(x,\tau)/\beta}}{Z_k(x)}, \qquad Z_k(x) = \mathbb{E}_{\tau \sim \pi_k(\cdot|x)} e^{R(x,\tau)/\beta}. \tag{C.1}$$

For binary rewards, let $p_k(x) = \Pr_{\pi_k}(R = 1 \mid x)$. For every prompt with $0 < p_k(x) < 1$,

$$p_k^*(x) = \frac{e^{1/\beta} p_k(x)}{1 - p_k(x) + e^{1/\beta} p_k(x)} > p_k(x), \qquad \frac{p_k^*(x)}{1 - p_k^*(x)} = e^{1/\beta} \frac{p_k(x)}{1 - p_k(x)}. \tag{C.2}$$

Repeated exact updates with fixed β make these success probabilities converge to one.

The corresponding objective-gap identity is

$$J_k(\pi_k^*) - J_k(\theta) = \beta \mathbb{E}_x D_{\mathrm{KL}}(\pi_\theta(\cdot \mid x) \| \pi_k^*(\cdot \mid x)) \,. \tag{C.3}$$

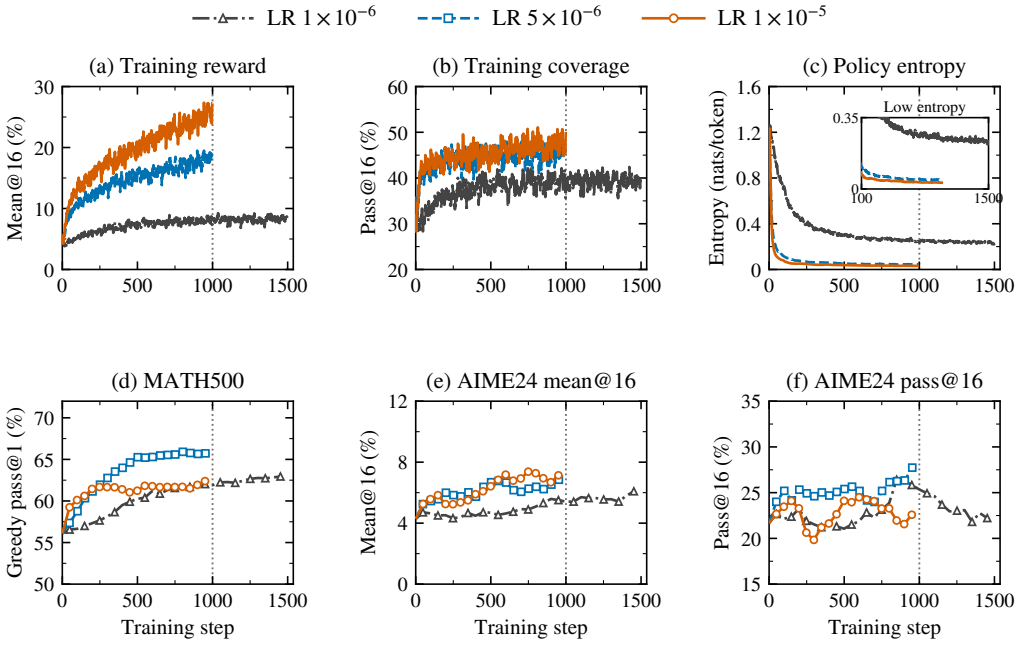

Figure 6: Qwen3-1.7B DAPO learning-rate sweep, lag 4. Dotted line: 1,000 updates; display cutoff: 1,500. Restart-adjacent evaluations use per-metric maxima before EMA $\alpha = 0.20$.

Proof of Proposition C.1. Substituting the Gibbs form into the KL divergence gives the objective-gap identity. Its right-hand side is nonnegative and vanishes only at $\pi_\theta = \pi_k^*$, proving maximality. For binary rewards, conditioning the Gibbs form on success gives Equation (C.2). Each exact reference refresh adds $1/\beta$ to the success log-odds, so repeated exact updates send the success probability to one. □

C.1 Illustration of proximal feedback with shared parameters

Proximal and trust-region methods have an established optimization motivation (Schulman et al., 2015; Tomar et al., 2020). The following construction illustrates how proximal feedback can affect interference between prompts, providing background for the on-policy comparison. It compares ordinary Euclidean reward-gradient flow with repeated exact proximal solves in the same policy class.

Construction. Take two equally likely prompts, a single action $a \in \{0, 1\}$, and reward $R = a$. Let $p_m = \pi_\theta(1 \mid x_m) = \sigma(u_m^\top \theta)$, where σ is the logistic function, and set

$$u_1 = (1, 1)^\top/5, \qquad u_2 = (1, -4)^\top/5, \qquad \theta_0 = (-\Delta, \Delta)^\top,$$
$$\Delta = \log \frac{1 - p_{\text{init}}}{p_{\text{init}}}, \qquad 0 < p_{\text{init}} \le 1/64.$$

Initially $(p_1, p_2) = (1/2, p_{\text{init}})$. The shared head can solve both prompts: along $\theta = (c, 0)^\top$, both success probabilities approach one as $c \to \infty$.

Proposition C.2 (Reward-gradient flow versus exact proximal updates)**.** *From this initialization, Euclidean gradient flow for $J(\theta) = (p_1 + p_2)/2$ converges in policy space to $(p_1, p_2) = (1, 0)$ and never achieves $J > 1/2$. Repeated exact proximal updates with any fixed $\beta > 0$ are representable by the same head and converge to $(p_1, p_2) = (1, 1)$.*

Proof. Write $z_m = u_m^\top \theta$ and $d_m = p_m(1 - p_m)$. Reward-gradient flow satisfies

$$\dot{z}_1 = (2d_1 - 3d_2)/50, \qquad \dot{z}_2 = (-3d_1 + 17d_2)/50.$$

In the region $z_1 \geq 0$, $z_1 + z_2 \leq -\Delta$, we have $d_1 \geq e^{-z_1}/4$ and $d_2 \leq e^{-z_1} p_{\text{init}}/(1 - p_{\text{init}})$, hence $d_2/d_1 \leq 4/63 < 1/14$. Thus $\dot{z}_1 > 0$ and $\dot{z}_1 + \dot{z}_2 = (-d_1 + 14d_2)/50 < 0$, so the region is invariant and contains the initialization. It implies $p_2 \leq 1 - p_1$ and $J \leq 1/2$. If the increasing z_1 were bounded, d_1 and consequently $\dot{z}_1$ would be bounded below by positive constants, a contradiction. Therefore $z_1 \to \infty$ and $z_2 \to -\infty$.

By Proposition C.1, an exact proximal update adds $1/\beta$ to each success logit. The parameter increment $(5, 0)^\top/\beta$ realizes both changes simultaneously. Repeating these exact solves gives

$$(p_{K,1}, p_{K,2}) = \big(\sigma(K/\beta), \sigma(-\Delta + K/\beta)\big) \longrightarrow (1, 1).$$

□

Proximal feedback. For a fixed reference with logits $z_{k,m}$, differentiation gives

$$\nabla J_k(\theta) = \frac{1}{2} \sum_{m=1}^{2} u_m p_m (1 - p_m)\{1 - \beta(z_m - z_{k,m})\}.$$

The negative inner product $u_1^\top u_2$ permits interference between prompts. The factor in braces decreases to zero as a prompt's logit increment reaches $1/\beta$, limiting continued domination by the initially easier prompt.

C.2 The Gradient of Proximal Improvement

FRPO identifies the regularization credit associated with each generated token. An earlier token influences later prefixes, so its coefficient includes subsequent regularization costs. Below we specialize its reward utility to a linear function and keep the reference fixed within one proximal round.

Take $\ell(p) = p$ and $f(u) = -\log u$ in Yao et al. (2026, Theorem 4.1, Equations (4.2)–(4.5)), with the fixed reference renamed $\pi_{\text{ref}} = \pi_k$. Their ratio is $u_i = 1/\rho_i$. For $h_i = (x, a_{<i})$, write $\ell_i = \log \rho_i$ and $\psi_i = \nabla_\theta \log \pi_\theta(a_i \mid h_i)$. For any history-only baseline $b(h_i)$, independent of the current action, the cited identity gives

$$\widehat{A}_i^{\text{KL}} = R - b(h_i) - \beta \sum_{j=i}^{T} \ell_j, \tag{C.4}$$

$$\nabla_\theta J_k(\theta) = \mathbb{E}_{\pi_\theta} \sum_{i=1}^{T} \psi_i \widehat{A}_i^{\text{KL}}. \tag{C.5}$$

The sampled coefficient is treated as a stop-gradient weight. Its sum includes the current token and later regularization costs; omitting the future costs generally changes the gradient. FRPO's Section 4.3 distinguishes this credit-assignment issue from stale-data correction. The latter is addressed in Appendix D.

D Asynchronous Updates with Token-Wise Importance Weighting

Asynchronous learning reuses responses from older policies. We analyze a PG update using sampled rewards, full future credit, and one weight per sampled token, without fitting a value function. The main result characterizes its stationary policy; convergence is then established under stronger population-flow assumptions.

D.1 Exact Response Correction

We first write the exact importance correction for a complete response. The independent-token example then shows why multiplying small token mismatches can produce a large sampling-weight variance. It motivates examining a cheaper token-wise importance correction.

Proposition D.1 (Exact response-level importance correction)**.** *Fix the learner, reference, and behavior policies on the finite generation tree, with the same prompt distribution and positive token*

probabilities. Let $\psi_i = \nabla_\theta \log \pi_\theta(a_i \mid h_i)$, and use the full future KL advantage from Equation (3) with a history-only baseline. Then

$$\frac{\pi_\theta(\tau \mid x)}{\pi_{\text{behav}}(\tau \mid x)} = \prod_{i=1}^{T} \left[\frac{\pi_{\text{prox}}(a_i \mid h_i)}{\pi_{\text{behav}}(a_i \mid h_i)} \rho_i \right], \qquad \nabla_\theta J_k(\theta) = \mathbb{E}_{\pi_{\text{behav}}} \left[\frac{\pi_\theta(\tau \mid x)}{\pi_{\text{behav}}(\tau \mid x)} \sum_i \psi_i \widehat{A}_i^{\text{KL}} \right]. \tag{D.1}$$

Proof. The autoregressive factorization gives the response likelihood ratio. Changing measure from the behavior policy to the learner policy then recovers the full future KL gradient in Equation (C.5). □

Sampling-weight variance.

Proposition D.2 (Variance of a product of token ratios)**.** *Let T binary tokens be independent, with behavior probability $1/2$ and learner probability $1/2 + \delta$ for the binary token value $a_i = 1$ at every position, where $0 < |\delta| < 1/2$. Write $W_{1:T}(\pi_\theta, \pi_{\text{behav}}) = \pi_\theta(\tau \mid x)/\pi_{\text{behav}}(\tau \mid x)$. Then*

$$\mathbb{E}_{\pi_{\text{behav}}}[W_{1:T}(\pi_\theta, \pi_{\text{behav}})^2] = (1 + 4\delta^2)^T, \qquad \text{Var}_{\pi_{\text{behav}}}(W_{1:T}(\pi_\theta, \pi_{\text{behav}})) = (1 + 4\delta^2)^T - 1. \tag{D.2}$$

Proof. Each token ratio equals $1 + 2\delta$ or $1 - 2\delta$ with equal behavior probability, so its mean is one and its second moment is $1 + 4\delta^2$. Independence gives the product's second moment; subtracting its squared mean gives the variance. □

This calculation quantifies the variance of response-level importance weights.

D.2 Token-Wise Importance Weighting and Its Stationary Target

The pathwise cancellation in Steps 1–2 is the soft-consistency mechanism of PCL and Trust-PCL (Nachum et al., 2017; 2018). We use it to characterize the zero of the PG field with token-wise importance weighting. Step 4 links its score moment to actionwise conditions and identifies the stationary policy despite stale continuations. The values in the proof are analytical quantities; the learner uses terminal rewards and full future credit. Here "synchronous" means $\pi_{\text{behav}} = \pi_\theta$ at each update, which is stronger than synchronous collection followed by rollout reuse. In Step 4, $q(h, a)$ abbreviates the previously defined $q_{\text{behav}}(h, a)$.

The reference-refresh schedule and actor-synchronization schedule are independent. Let M be the number of learner updates before refreshing the reference and B the response batch size. The analysis uses unnormalized rewards and the full Future-KL tail. The baseline $b(h_i)$ depends on the history but not on the sampled action and is detached during the update.

Algorithm. Initialize the learner at θ_0 and publish its policy to the actors.

1. **Collect responses.** Each actor generates with its available learner snapshot. Store the response, reward, and the actual sampling probability $\pi_{\text{behav}}(a_i \mid h_i)$ for every token.
2. **Freeze the reference.** At the start of round k, set $\pi_{\text{prox}} = \pi_k = \pi_{\theta_{k,0}}$ to the current learner policy. Keep this reference fixed for the next M learner updates.
3. **Update the learner.** For $n = 0, \ldots, M - 1$, take B queued responses. Recompute each token probability under the current learner and the fixed reference, and form

$$\widehat{A}_i^{\text{KL}} = R - b(h_i) - \beta \sum_{j=i}^{T} \log \rho_j. \tag{D.3}$$

Take the ascent step

$$\widehat{g}_{k,n} = \frac{1}{B} \sum_{(x,\tau)\in\mathcal{B}_{k,n}} \sum_{i=1}^{T} \frac{\pi_{\text{prox}}(a_i \mid h_i)}{\pi_{\text{behav}}(a_i \mid h_i)} \rho_i \widehat{A}_i^{\text{KL}} \nabla_\theta \log \pi_\theta(a_i \mid h_i), \tag{D.4}$$

$$\theta_{k,n+1} = \theta_{k,n} + \alpha_{k,n} \widehat{g}_{k,n}. \tag{D.5}$$

Recompute the learner-dependent terms after every parameter update.

4. **Refresh and repeat.** Set $\theta_{k+1,0} = \theta_{k,M}$ and $\pi_{k+1} = \pi_{\theta_{k,M}}$. Return to the reference-freezing step; asynchronous collection continues throughout.

Implementation rule. The behavior denominator remains the probability logged by the actor that generated the sample. If a queued response crosses a reference-refresh boundary, keep its stored behavior probabilities and recompute only the reference and learner probabilities. A batch may therefore contain several behavior versions.

An equivalent ascent surrogate for (D.4) is

$$\widehat{\mathcal{L}}_{k,n}(\theta) = \frac{1}{B} \sum_{(x,\tau)\in\mathcal{B}_{k,n}} \sum_{i=1}^{T} \frac{\pi_{\text{prox}}(a_i \mid h_i)}{\pi_{\text{behav}}(a_i \mid h_i)} \rho_i(\theta) \, \text{sg}[\widehat{A}_i^{\text{KL}}], \tag{D.6}$$

where sg stops gradients. Differentiating the token ratio produces the score term in (D.4). The practical benefit is visible here: the update uses one explicit $\pi_{\text{prox}}/\pi_{\text{behav}}$ correction per token and never multiplies behavior corrections over the response.

D.2.1 Assumptions for Theorem 1

(A1) Finite supported model. The prompt distribution has finite support with positive probability on every included prompt. Each prompt has a finite token-generation tree and a fixed terminal reward. The reference π_k is fixed during the round, and both π_k and every learner policy under consideration assign positive probability to every allowed token.

(A2) Covered behavior sampling. Responses come from a fixed behavior policy or a mixture of actor versions. Each response retains the actual version-specific token probabilities used to sample it. Every nonterminal history in the generation tree has positive behavior occupancy; all such histories are relevant because the reference has full support. Every contributing version assigns positive probability to each allowed next token at that history.

(A3) Representability and local expressiveness. The learner class contains π_k^*. Collect all relevant independent next-token log odds into $z(\theta)$. Its Jacobian $Dz(\theta)$ has full row rank at every parameter considered, so local parameter changes span every such log-odds direction. Independent tabular logits satisfy this condition.

(A4) Ideal population update. The field is the exact expectation of Equation (D.7) under the behavior sampling law, held fixed during differentiation; the conditional expectations defining q_{behav} are exact population quantities. The field uses the full, unclipped future KL sum and an action-independent, history-only baseline held fixed during differentiation. No finite-horizon truncation, ratio clipping, group normalization, or finite-batch approximation enters this field.

We index context depth by the next token position, so $\text{depth}(h_i) = i$.

The token-wise expected update is

$$M_{\text{behav}}(\theta) = \mathbb{E}_{\pi_{\text{behav}}} \sum_{i=1}^{T} \frac{\pi_{\text{prox}}(a_i \mid h_i)}{\pi_{\text{behav}}(a_i \mid h_i)} \rho_i \psi_i \widehat{A}_i^{\text{KL}}. \tag{D.7}$$

Proof of Theorem 1. For a history h, define the soft partition function, value, and action value by

$$\begin{aligned} Z_k(h) &= \mathbb{E}_{\pi_k(\cdot|h)} e^{R/\beta}, & V_k(h) &= \beta \log Z_k(h), \\ Q_k(h,a) &= V_k(ha), & V_k(h_{T+1}) &= R. \end{aligned} \tag{D.8}$$

Step 1: obtain the token-level proximal policy. The partition function at context h averages the partition functions of its children:

$$Z_k(h) = \sum_a \pi_k(a \mid h) Z_k(ha). \tag{D.9}$$

Conditioning the response-level Gibbs law (C.1) on context h therefore gives

$$\pi_k^*(a \mid h) = \pi_k(a \mid h) \frac{Z_k(ha)}{Z_k(h)}. \tag{D.10}$$

Taking logs and using $V_k = \beta \log Z_k$ gives the regularized Q-function identity

$$\log \frac{\pi_k^*(a \mid h)}{\pi_k(a \mid h)} = \frac{Q_k(h,a) - V_k(h)}{\beta}. \tag{D.11}$$

Step 2: telescope along a sampled suffix. Define the future-only target and rewrite the algorithm's advantage as

$$\widehat{Q}_i^{\pi_\theta} = R - \beta \sum_{j>i} \ell_j, \qquad \widehat{A}_i^{\mathrm{KL}} = \widehat{Q}_i^{\pi_\theta} - \beta \ell_i - b(h_i). \tag{D.12}$$

Apply (D.11) to every token after i:

$$\begin{aligned} \beta \sum_{j>i} \log \frac{\pi_k^*(a_j \mid h_j)}{\pi_k(a_j \mid h_j)} &= \sum_{j>i} \{V_k(h_{j+1}) - V_k(h_j)\} \\ &= R - V_k(h_{i+1}). \end{aligned} \tag{D.13}$$

All intermediate values cancel. Since $h_{i+1} = h_i a_i$, the remaining value is $V_k(h_{i+1}) = Q_k(h_i, a_i)$. Substitution into (D.12) proves the pathwise identities

$$\widehat{Q}_i^{\pi_k^*} = Q_k(h_i, a_i), \qquad \widehat{A}_i^{\mathrm{KL}}(\pi_k^*, \tau) = V_k(h_i) - b(h_i). \tag{D.14}$$

They hold for every supported suffix, before taking any expectation.

Step 3: measure the error away from the solution. Subtract the same telescoping identity for π_k^* from the target formed with an arbitrary learner π_θ. This gives

$$\widehat{Q}_i^{\pi_\theta} - Q_k(h_i, a_i) = -\beta \sum_{j>i} \log \frac{\pi_\theta(a_j \mid h_j)}{\pi_k^*(a_j \mid h_j)}. \tag{D.15}$$

The target error at token i is exactly the sum of the downstream log-policy errors, scaled by $-\beta$. This identity also controls the perturbation term in the convergence proof below.

Step 4: identify the zero of the expected update. Let $d_{\mathrm{behav}}(h)$ include both prompt probability and behavior prefix occupancy. Write $\psi_\theta(h,a) = \nabla \log \pi_\theta(a \mid h)$. To include mixed actor versions, let v denote the version that generated a response and let $\omega(v \mid h)$ be its conditional probability given the observed context, before sampling the next action. For $i = \mathrm{depth}(h)$, define

$$q_{\mathrm{behav}}(h,a) = \sum_v \omega(v \mid h) \mathbb{E}_{\pi_{\mathrm{behav},v}}[\widehat{Q}_i^{\pi_\theta} \mid h_i = h, a_i = a]. \tag{D.16}$$

This samples a version conditional on h, fixes the current action to a, and then samples its continuation. The token ratio cancels that version's current-action probability, so the mixture weights are $\omega(v \mid h)$, not the weights conditional on both h and a. A single behavior policy is the special case of one version. Taking the expectation of the token-wise policy-gradient estimator in (D.7), including the logged version of each sample, and applying current-action importance sampling gives the expected PG field

$$\begin{aligned} M_{\mathrm{behav}}(\theta) = \sum_h d_{\mathrm{behav}}(h) \sum_a & \pi_\theta(a \mid h) \psi_\theta(h,a) \\ & \left[q_{\mathrm{behav}}(h,a) - \beta \log \frac{\pi_\theta(a \mid h)}{\pi_k(a \mid h)} - b(h) \right]. \end{aligned} \tag{D.17}$$

In synchronous training, this field equals $\nabla J_k(\theta)$. With stale continuations, it is the expected update proposed in (D.7), but it is generally not the exact gradient of J_k; obtaining that gradient requires the sequence-wise correction in (D.1). The importance ratio corrects only the probability of the current action. The probabilities of reaching the history and of sampling the later tokens remain those of the behavior policy. For tabular logits at h, define $u(h,a) = q_{\mathrm{behav}}(h,a) - \beta \log(\pi_\theta(a \mid h)/\pi_k(a \mid h))$. Theorem D.3 below derives the corresponding ordinary-PG logit block and its Lyapunov identity. Here we only need its zero condition: with independent logits and positive behavior occupancy, the expected block vanishes exactly when $u(h,a)$ is constant over actions. A softmax has a redundant common-shift direction; passing from its full logit vector to independent log-odds removes only that direction and leaves this zero condition unchanged. At the last token, $q(h,a) = R = Q_k(h,a)$, so this condition is equivalent to (D.11); hence the last-token policy must be π_k^*. If all later-token policies equal π_k^*, Step 2 makes $q(h,a) = Q_k(h,a)$ at the current history, and the same argument forces its policy to equal π_k^*. Backward induction proves that the token-wise field vanishes at every history exactly at the proximal policy. For a shared-parameter model, collect the independent log odds into a vector $z(\theta)$ and let g_z be this same field in those coordinates. The chain rule gives $M_{\mathrm{behav}}(\theta) = Dz(\theta)^\top g_z$. Full row rank of $Dz(\theta)$ makes its transpose injective, so the parameter-space field is zero exactly when $g_z = 0$. This proves Theorem 1 under the stated local controllability assumption. □

D.3 Population Convergence and Outer Improvement

This section gives one sufficient example of periodic reference refresh with approximate inner solves. We establish convergence of the tabular population flow and then show that a success-log-odds tolerance yields improvement after each refresh. The example shows how an explicit inner-accuracy condition supports progress across rounds.

For the convergence result, we specialize Theorem 1 to independent softmax logits at every history, continuous-time population updates, and behavior occupancy uniformly bounded away from zero. These conditions yield a direct Lyapunov analysis of the update dynamics.

Fix π_k and follow this ordinary population PG field. Let t denote time, let $d_t(h)$ be the behavior occupancy of history h, and define, for $i = \mathrm{depth}(h)$,

$$q_t(h,a) = \sum_v \omega_t(v \mid h)\mathbb{E}_{\pi_{\mathrm{behav},t,v}}[\widehat{Q}_i^{\pi_{\theta(t)}} \mid h_i = h, a_i = a]. \tag{D.18}$$

Theorem D.3 (Convergence of the asynchronous population flow)**.** *Assume (A1)–(A4), with independent softmax logits at every nonterminal history and a fixed reference π_k. Allow the behavior policy or actor-version mixture to vary measurably with time, with the corresponding sampling law held fixed when forming each update. Follow the continuous-time population logit flow from finite initial logits. Suppose $d_t(h) \geq d_{\min} > 0$ for every nonterminal history and all $t \geq 0$. Fix a history h with m allowed actions. Let $\boldsymbol{p}_h(t) = \pi_{\theta(t)}(\cdot \mid h)$, $\boldsymbol{p}_h^* = \pi_k^*(\cdot \mid h)$, and*

$$P_h = I - \frac{1}{m}\mathbf{1}\mathbf{1}^\top, \qquad \boldsymbol{r}_h(t) = P_h \log \frac{\boldsymbol{p}_h(t)}{\boldsymbol{p}_h^*}.$$

The logarithm and division are componentwise. Write $\boldsymbol{e}_h(t) = q_t(h,\cdot) - Q_k(h,\cdot)$ and $F(\boldsymbol{p}) = \mathrm{diag}(\boldsymbol{p}) - \boldsymbol{p}\boldsymbol{p}^\top$. Then the ordinary expected PG logit flow at h is

$$\dot{\boldsymbol{z}}_h(t) = d_t(h)F(\boldsymbol{p}_h(t))\{-\beta \boldsymbol{r}_h(t) + \boldsymbol{e}_h(t)\}. \tag{D.19}$$

Consequently, for $V_h(t) = \frac{1}{2}\|\boldsymbol{r}_h(t)\|_2^2$,

$$\begin{aligned}\dot{V}_h(t) = &-\beta d_t(h)\boldsymbol{r}_h(t)^\top F(\boldsymbol{p}_h(t))\boldsymbol{r}_h(t) \\ &+ d_t(h)\boldsymbol{r}_h(t)^\top F(\boldsymbol{p}_h(t))\boldsymbol{e}_h(t).\end{aligned} \tag{D.20}$$

The policies satisfy $\boldsymbol{p}_h(t) \to \boldsymbol{p}_h^*$ at every history, including under arbitrary time-varying supported behavior mixtures satisfying the occupancy bound.

Proof. Let $\boldsymbol{e}_a$ be the one-hot vector for action a. The softmax identity $\nabla_{\boldsymbol{z}_h} \log p_h(a) = \boldsymbol{e}_a - \boldsymbol{p}_h$ and (D.17) give the ordinary PG logit block

$$d_t(h)F(\boldsymbol{p}_h)\left\{q_t(h,\cdot) - \beta \log \frac{\boldsymbol{p}_h}{\pi_k(\cdot \mid h)}\right\};$$

the action-independent baseline vanishes because $F(\boldsymbol{p}_h)\mathbf{1} = \mathbf{0}$. The proximal identity (D.11) implies

$$Q_k(h,\cdot) - \beta \log \frac{\boldsymbol{p}_h^*}{\pi_k(\cdot \mid h)} = V_k(h)\mathbf{1}.$$

Subtracting this identity inside the PG block and again removing constants with $F(\boldsymbol{p}_h)\mathbf{1} = \mathbf{0}$ proves (D.19). Finally, $\boldsymbol{r}_h = P_h(\boldsymbol{z}_h - \boldsymbol{z}_h^*)$ and the right-hand side of (D.19) sums to zero, so $\dot{\boldsymbol{r}}_h = \dot{\boldsymbol{z}}_h$. Taking the inner product with $\boldsymbol{r}_h$ proves (D.20).

Backward convergence of ordinary PG. At the last token position, (D.14) gives $\boldsymbol{e}_h(t) = \mathbf{0}$. Equation (D.20) therefore makes $V_h(t)$ nonincreasing. The centered log ratios remain bounded, so normalization and the positive support of $\boldsymbol{p}_h^*$ keep every component of $\boldsymbol{p}_h(t)$ uniformly away from zero. Consequently, $\lambda_h := \inf_{t\geq 0} \min_a p_h(t,a) > 0$. For centered $\boldsymbol{r}_h$, the weighted-variance identity gives

$$\boldsymbol{r}_h^\top F(\boldsymbol{p}_h)\boldsymbol{r}_h = \mathrm{Var}_{a\sim \boldsymbol{p}_h}[r_h(a)] \geq \lambda_h \|\boldsymbol{r}_h\|_2^2.$$

Thus $\dot{V}_h \leq -2\beta d_{\min}\lambda_h V_h$, and the last-token policies converge exponentially to $\boldsymbol{p}_h^*$.

Proceed backward by induction. Define

$$\delta_j(t) = \max_{h:\,\mathrm{depth}(h)=j} \mathrm{osc}_a \log \frac{\pi_{\theta(t)}(a \mid h)}{\pi_k^*(a \mid h)}.$$

Because both policies are normalized, their log probability ratio either vanishes for every action or takes both nonpositive and nonnegative values. Its absolute value is therefore bounded by its oscillation $\delta_j(t)$. If all depths after i converge exponentially, the tail identity (D.15) gives, uniformly over histories at depth i,

$$\|\boldsymbol{e}_h(t)\|_\infty \leq \beta \sum_{j>i} \delta_j(t) \leq C_i e^{-c_i t}$$

for some $C_i, c_i > 0$. To handle $\boldsymbol{r}_h = \mathbf{0}$ without a separate case, fix $\eta > 0$. The flow equation and the negative semidefiniteness of its first term imply

$$\frac{d}{dt}\sqrt{\|\boldsymbol{r}_h(t)\|_2^2 + \eta} \leq d_t(h)\|F(\boldsymbol{p}_h(t))\|_{\mathrm{op}}\|\boldsymbol{e}_h(t)\|_2.$$

The right-hand side is integrable. Integrating and then letting $\eta \downarrow 0$ shows that $\boldsymbol{r}_h(t)$ remains bounded, so $\boldsymbol{p}_h(t)$ again remains uniformly in the interior. Applying Young's inequality to (D.20) then gives constants $c_h, C_h > 0$ such that

$$\dot{V}_h(t) \leq -c_h V_h(t) + C_h\|\boldsymbol{e}_h(t)\|_2^2.$$

The forcing term decays exponentially, so Gronwall's inequality yields $V_h(t) \to 0$ exponentially. Backward induction proves convergence to π_k^* at every history. □

Corollary D.4 (Reference refresh with approximate inner solves)**.** *Assume a finite prompt set, binary reward, and a full-support initial reference with $0 < p_0(x) < 1$ for every prompt, where $p_k(x) = \Pr_{\tau\sim\pi_k(\cdot|x)}[R(x,\tau) = 1]$. Fix $\beta > 0$ and $\varepsilon_{\mathrm{logit}} \in (0, 1/\beta)$. At round k, let π_k^* maximize the proximal objective with reference π_k, write $p_k^*(x) = \Pr_{\tau\sim\pi_k^*(\cdot|x)}[R(x,\tau) = 1]$, and refresh the reference to a full-support approximation π_{k+1} satisfying, for every prompt,*

$$|\mathrm{logit}\, p_{k+1}(x) - \mathrm{logit}\, p_k^*(x)| \leq \varepsilon_{\mathrm{logit}}. \tag{D.21}$$

Then, with $\mathrm{logit}(p) = \log(p/(1-p))$ and $\sigma(u) = (1+e^{-u})^{-1}$,

$$p_k(x) \geq \sigma(\mathrm{logit}\, p_0(x) + k(1/\beta - \varepsilon_{\mathrm{logit}})) \longrightarrow 1.$$

Under the assumptions of Theorem D.3 in each round, its inner flow reaches the required tolerance in finite time.

Proof. By Proposition C.1,

$$\mathrm{logit}\, p_k^*(x) = \mathrm{logit}\, p_k(x) + 1/\beta.$$

Hence each approximate round increases every success log-odds by at least $1/\beta - \varepsilon_{\mathrm{logit}} > 0$. Summing these increments and applying σ proves the bound and convergence. Theorem D.3 gives convergence to π_k^* in each inner flow. The finite prompt set and continuity of success log-odds then give a finite stopping time satisfying Equation (D.21) for all prompts. □

Here a finite inner solve is measured directly by its log-odds distance from the proximal target. This formulation separates inner optimization accuracy from the exact-solve improvement in Proposition C.1 and shows that any fixed tolerance below $1/\beta$ preserves positive progress at every outer round. More general inner-accuracy conditions are a natural direction for future work.

E HYPERPARAMETERS AND COMPUTE

Table 5 groups the trained main-results settings by method; merged cells indicate shared values. Here H counts action tokens and β weights each retained future cost. Both models use the 1,000-update evaluation budget (Section 5.1). All runs use Adam with $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon = 1 \times 10^{-8}$, a constant learning rate, no warmup, and no weight decay. The 4B *Proximal KL* main run uses $H = 8$ and $\beta = 0.25$.

Hardware. Each main run uses four nodes with eight H100 80GB GPUs per node (32 GPUs). Training uses bfloat16 and gradient checkpointing.

Table 5: Hyperparameters for Table 1. Merged cells denote shared values.

Method	Setting	Off-policy			On-policy
	Model	Qwen3-1.7B		Qwen3-4B	Qwen3-1.7B
	Learning rate	1×10^{-5}	5×10^{-6}	5×10^{-6}	5×10^{-6}
DAPO	PPO clip	Yes			No
	Prompts per batch	128			
	Responses per prompt	16			
	Response limit (tokens)	8,192			
	Sampling temperature	1.0			
	Optimizer	Adam			
	Gradient-norm limit	1.0			
	Minibatches per batch	4			1
	Reward normalization	Within each prompt group			
	Offpolicyness cap	4			0
	Training seeds	1	3	1	
	Evaluation budget (updates)	1,000			
Proximal KL	Future horizon H	Full		8	Full
	Coefficient β	0.30		0.25	0.01
	PPO clip	No			
	Future log-ratio bound ϵ_F	0.05			No clipping
	Reference refresh (updates)	1			10
	Prompts per batch	128			
	Responses per prompt	16			
	Response limit (tokens)	8,192			
	Sampling temperature	1.0			
	Optimizer	Adam			
	Gradient-norm limit	1.0			
	Minibatches per batch	4			1
	Reward normalization	Within each prompt group			
	Offpolicyness cap	4			0
	Training seeds	1	3	1	
	Evaluation budget (updates)	1,000			