FUTURE-KL REGULARIZED GRPO: PROCESS-LEVEL CREDIT ASSIGNMENT FROM f-DIVERGENCE REGULARIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Group Relative Policy Optimization (GRPO) is widely used for critic-free Large Language Model (LLM) post-training, but its KL regularization is usually implemented as a local loss-side token penalty. We show that this misses the policy-gradient signal induced by autoregressive KL regularization. Unlike standard KL-regularized Reinforcement Learning (RL) objectives, GRPO's group normalization induces a non-linear prompt-level utility; for binary verifier rewards, this utility is $2\arcsin\sqrt{p}$. As a result, reward and KL cannot be fused before normalization without changing the implicit objective. We derive the on-policy gradient of GRPO-style objectives with token-wise f-divergence regularization. The reward term recovers the standardized GRPO advantage, while the regularizer contains a causal future cost-to-go omitted by local KL losses. For reverse KL, this yields **F**uture-KL **R**egularized **P**olicy **O**ptimization (**FRPO**), a critic-free correction implemented by a reverse cumulative sum of sampled token log-ratios after advantage construction. We further study a windowed variant that interpolates between local and full-horizon regularization credit. On mathematical-reasoning tasks, full-horizon FRPO improves pass@16 in the experiments, maintaining a better exploration ability compared to other regularization methods.

1 INTRODUCTION

RL has become a central component of post-training LLMs for mathematical reasoning and other verifiable tasks. A widely used paradigm is to sample multiple responses for each prompt, score them using an outcome verifier, and update the policy toward responses that outperform others in the same group. GRPO (Shao et al., 2024; Guo et al., 2025) follows this paradigm and removes the need for a learned critic by normalizing outcome rewards within each prompt group. This critic-free design has made GRPO and its variants attractive for large-scale reasoning training, where outcome rewards are often sparse, binary, and inexpensive to verify.

A key ingredient in such training pipelines is KL regularization against a reference policy. KL regularization stabilizes policy updates, prevents excessive drift from the supervised model, and preserves general capabilities while optimizing for task-specific rewards. In the original GRPO formulation, the KL penalty is added as a token-level loss-side regularizer, often using a sampled estimator such as the $k3$ estimator (Schulman, 2020). This implementation is simple and empirically useful, but it obscures several distinct questions: which sampling distribution the sampled KL term estimates under, whether its gradient corresponds to the intended KL objective, and, more importantly, how KL regularization should assign credit across tokens in an autoregressive trajectory.

Recent analyses (Zhang et al., 2025b) of KL-regularized policy-gradient methods have clarified several implementation-level issues, including stale-rollout distribution mismatch and the gap between value-unbiased and gradient-unbiased sampled KL estimators. These observations are important, but they do not fully answer a more basic on-policy question: what token-level policy-gradient signal is induced by regularization itself when the expectation is over the current autoregressive policy?

We therefore separate two issues that are often conflated. The first is an estimator issue: rollouts may be sampled from $\pi_{\theta_{\text{old}}}$ while a KL term is evaluated under π_θ, requiring importance correction or fresh log-probabilities. The second is an objective issue: even when samples are on-policy, a

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

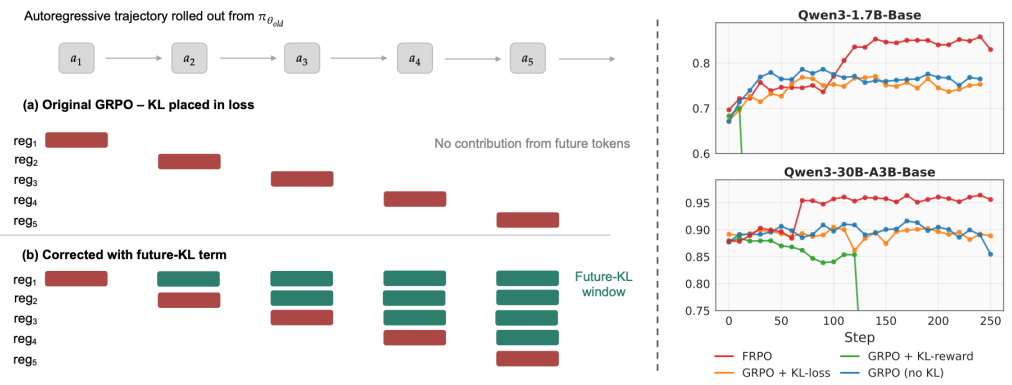


Figure 1: **Left:** Accumulating future KL divergence between the current policy π_θ and the reference policy π_{ref} contributes to the trajectory-level regularization signal, but is not captured by an independent loss-side token penalty. **Right:** Pass@16 dynamics of Qwen3-1.7B-Base and Qwen3-30B-A3B-Base under different KL integrations.

loss-side local KL term differentiates only the sampled token penalty and ignores how the current token changes the distribution of future states. This paper focuses on the second issue and shows that it leads to a different token-level credit-assignment rule.

This issue is fundamental in autoregressive language modeling. An early token not only incurs its own local regularization cost, but also changes the future prefix distribution and hence the regularization costs of later tokens. Consequently, token-wise f-divergence regularization induces a process-level credit assignment signal. For reverse KL, the policy-gradient weight for each token should include a future-KL cost-to-go term, rather than only an independent token-wise loss penalty.

Why GRPO differs from standard MaxEnt RL. Standard KL-regularized RL uses an objective linear in reward, allowing reward and regularization to be combined into one augmented reward. GRPO instead group-normalizes outcome rewards, inducing a non-linear prompt utility; reward and KL therefore receive different scaling, so the regularization correction belongs after advantage construction rather than inside the normalized reward. Section 3 formalizes this distinction.

We derive the on-policy gradient of a GRPO-style objective under this general non-linear-ℓ + token-wise-f structure, and show that the resulting token-level weight decomposes into an outcome-level group-relative advantage and a future regularization correction. This leads to a decoupled future-KL formulation for reverse KL: compute the GRPO advantage from the original verifier reward, then add token-dependent future-KL credit without modifying the group statistics. A finite horizon provides a tunable approximation between current-token and full-horizon credit. This formulation separates trajectory-level ranking from process-level KL credit assignment and differs from local loss-side regularization, KL-in-reward normalization, and actor-critic GAE.

In summary, this work makes the following contributions:

1. We show that population GRPO with binary verifier rewards follows a non-linear arcsine utility. Reward and regularization therefore receive different prompt-dependent scaling, so inserting KL before group normalization does not recover the stated regularized objective.
2. We derive the on-policy gradient for token-wise f-divergence regularization. The regularization coefficient at a token contains a causal future cost-to-go that local loss-side penalties omit.
3. We propose FRPO, which preserves GRPO's outcome advantage and adds token-level future-KL credit after advantage construction, together with a windowed variant interpolating between local and full-horizon credit. Our experimental results show the importance of incorporating the appropriate regularization into the optimization objective, where FRPO achieves better performance compared to other baselines.

2 PRELIMINARIES AND NOTATIONS

Consider an autoregressive language model π_θ generating a response $o = (o_1, o_2, \ldots, o_T)$ given a prompt x, θ represents the model's parameters. We write the conditional probability of generating response o from the prompt x as $\pi_\theta(o \mid x) = \prod_{t=1}^{T} \pi_\theta(o_t \mid x, o_{<t})$.

Let π_{ref} denote a frozen reference policy, and $\pi_{\theta_{\text{old}}}$ denote the policy at the time of rollout (i.e., the sampling policy) which is an older version of the current policy π_θ.

We define the following shorthand for token-level quantities at position t:

$$\delta_t \triangleq \log \pi_\theta(o_t \mid x, o_{<t}) - \log \pi_{\text{ref}}(o_t \mid x, o_{<t}) = \log \frac{\pi_\theta(o_t|x, o_{<t})}{\pi_{\text{ref}}(o_t|x_t, o_{<t})},$$
$$\alpha_t \triangleq \frac{\pi_\theta(o_t \mid x, o_{<t})}{\pi_{\text{ref}}(o_t \mid x, o_{<t})} = e^{\delta_t}, \quad u_t \triangleq \alpha_t^{-1}, \quad \rho_t \triangleq \frac{\pi_\theta(o_t \mid x, o_{<t})}{\pi_{\theta_{\text{old}}}(o_t \mid x, o_{<t})}.$$

The true token-level reverse KL divergence at position t (given context $s_t = (x, o_{<t})$) is:

$$D_{\text{KL}}\big[\pi_\theta(\cdot \mid s_t) \,\|\, \pi_{\text{ref}}(\cdot \mid s_t)\big] = \sum_{a \in \mathcal{V}} \pi_\theta(a \mid s_t) \log \frac{\pi_\theta(a \mid s_t)}{\pi_{\text{ref}}(a \mid s_t)}, \tag{1}$$

where $\mathcal{V}$ is the vocabulary. Here we slightly abuse the notations with both a and o representing the actions taken by the policy π.

KL Estimators Schulman (2020) proposes three widely adopted KL estimators, $\hat{D}^{k1} = \log\, \alpha$, $\hat{D}^{k2} = (\log \alpha)^2/2$, and $\hat{D}^{k3} = 1/\alpha + \log \alpha - 1$, for estimating reverse KL $D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}})$, among which $k3$ is used most commonly due to its unbiased value and nonnegative property. Appendix A analyzes their expectations and gradients. The corresponding forward KL estimators could be defined similarly, with the ratio α being simply reversed. While these estimators motivate the main examples, our derivation in Section 4 applies to general token-wise f-divergence generators.

3 CLASSICAL GRPO AND THE REGULARIZATION QUESTION

3.1 GRPO OBJECTIVE AND GROUP-NORMALIZED ADVANTAGE

DeepSeek-Math (Shao et al., 2024) proposes GRPO as a critic-free policy optimization method. For each prompt x, it samples a group of G responses $\{o_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot \mid x)$, evaluates scalar outcome rewards $r_i = r(x, o_i)$, and optimizes a PPO-style surrogate:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}\Big[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\Big\{\min\Big[\rho_{i,t}\hat{A}_{i,t}, \text{clip}(\rho_{i,t}, 1-\varepsilon, 1+\varepsilon)\hat{A}_{i,t}\Big] - \beta\,\hat{D}_{i,t}\Big\}\Big], \tag{2}$$

where $\rho_{i,t}$ is the ratio ρ_t of Section 2 for response o_i and $\hat{D}_{i,t}$ is a sampled token-level KL penalty against the reference policy. Equation (2) uses the customary per-response presentation. Our experiments use token-mean loss aggregation over all valid response tokens in the actor mini-batch (Appendix F).

Unlike PPO with a learned value function, GRPO constructs advantages by normalizing rewards within the prompt group:

$$\mu_G = \frac{1}{G}\sum_{i=1}^{G} r_i, \qquad \sigma_G = \sqrt{\frac{1}{G}\sum_{i=1}^{G}(r_i - \mu_G)^2},$$

and assigns every token in response i the same outcome advantage

$$\hat{A}_{i,t} = \hat{A}_i^{\text{GRPO}} = \frac{r_i - \mu_G}{\sigma_G}, \qquad t = 1, \ldots, |o_i|. \tag{3}$$

This design removes the value model but also means that GRPO's reward signal is trajectory-level: before any KL term is added, all tokens in the same response receive the same credit.

3.2 THE IMPLICIT NON-LINEAR UTILITY FROM GROUP NORMALIZATION

Before discussing KL regularization, we record a consequence of group normalization that is the structural source of every novelty in this paper. The arcsin objective induced by standardized binary rewards has also been noted in REINFORCE-Ada (Xiong et al., 2025b). We revisit it here not as a new policy-gradient observation, but to expose a GRPO-specific complication: in binary-reward reasoning tasks, the normalization in Eq. (3) is not a neutral preprocessing step; it determines a *non-linear* prompt-level utility followed by the population GRPO gradient. Standard derivations of KL-regularized policy gradients (soft-Q, MaxEnt RL, KL-in-reward PPO) target objectives that are *linear* in reward, $\mathbb{E}_\pi[r] - \beta\,\mathbb{E}_\pi[D_{\mathrm{KL}}]$, where reward and regularizer fuse into a single augmented reward. GRPO breaks that linearity, and as we show below, this is why putting KL into the reward before normalization changes the implicit objective rather than merely rescaling it.

Proposition 1 (Implicit objective of binary GRPO)**.** *Fix a prompt x with binary reward $r(x,a) \in \{0,1\}$ and pass probability $p_\theta(x) = \mathbb{E}_{a\sim\pi_\theta(\cdot|x)}[r(x,a)]$. Replacing finite-group statistics with their population values, the expected GRPO policy gradient for this prompt is*

$$\mathbb{E}_{a\sim\pi_\theta(\cdot|x)}\left[\frac{r(x,a)-p_\theta(x)}{\sqrt{p_\theta(x)(1-p_\theta(x))}}\nabla_\theta \log \pi_\theta(a|x)\right] = \nabla_\theta\left[2\arcsin\sqrt{p_\theta(x)}\right].$$

The proof is in Appendix B. Group normalization should therefore be viewed as part of the objective, not merely as a numerical rescaling: the population-standardized GRPO advantage gives the gradient of $\ell(p) = 2\arcsin\sqrt{p}$, with $\ell'(p) = 1/\sqrt{p(1-p)}$. Folding KL into the reward places it inside the normalization, changing the relative reward and KL weights from those in $\ell(p) - \beta K$.

Proposition 2 (Population direction of KL-in-reward normalization)**.** *Fix a prompt and let $D_\theta(a) = \log(\pi_\theta(a)/\pi_{\mathrm{ref}}(a)) = \sum_j \delta_j$, $p_\theta = \mathbb{E}_{\pi_\theta} r$, and $K_\theta = \mathbb{E}_{\pi_\theta} D_\theta$. For $\tilde{r} = r - \beta D_\theta$ with population standard deviation $\tilde{\sigma} > 0$, the expected score update from the standardized shaped reward is*

$$\mathbb{E}_{\pi_\theta}\left[\frac{\tilde{r} - \mathbb{E}_{\pi_\theta}\tilde{r}}{\tilde{\sigma}}\nabla_\theta \log \pi_\theta(a)\right] = \frac{\nabla_\theta p_\theta - \beta \nabla_\theta K_\theta}{\tilde{\sigma}}.$$

Thus it rescales the linear reward–KL direction, whereas the separately regularized GRPO target has gradient $\ell'(p_\theta)\nabla_\theta p_\theta - \beta\nabla_\theta K_\theta$ for $\ell(p) = 2\arcsin\sqrt{p}$.

GRPO's outcome advantage should continue to rank responses by verifier reward, while KL regularization should provide a separate token-level credit-assignment signal — this is the structural reason behind the general regularized objective in Eq. (5), and the "KL-in-reward collapse" observed in Section 6 is consistent with this distortion. Proposition 2 gives the population direction; finite-group normalization adds the following concrete mechanism for the observed collapse. In any group whose verifier rewards are all equal and whose KL values vary, the outcome advantage is zero, but the KL-shaped advantage becomes $-(D_i - \bar{D})/\mathrm{Std}(D)$, a unit-variance signal driven purely by within-group KL variation in which β cancels entirely (Appendix C.1). With small groups and binary rewards, such groups are common, so KL-in-reward injects an $O(1)$ KL-driven update on a large fraction of prompts regardless of how small β is (about three quarters of groups in our 1.7B runs; Appendix F.6). Appendix E.1 discusses the alternative placements in detail.

3.3 LOSS-SIDE KL IN THE CLASSICAL FORMULATION

The KL term commonly used in GRPO implementations is the sampled $k3$ estimator (Schulman, 2020). With $u_t = \pi_{\mathrm{ref}}(o_t \mid s_t)/\pi_\theta(o_t \mid s_t) = \alpha_t^{-1}$, it is

$$\hat{D}_t^{(k3)}(\pi_\theta\|\pi_{\mathrm{ref}}) = u_t - \log u_t - 1 = \frac{1}{\alpha_t} - \log\frac{1}{\alpha_t} - 1. \tag{4}$$

Proposition 3 (Value unbiasedness under on-policy sampling)**.** *For a fixed state s_t, $\hat{D}_t^{(k3)}$ is an unbiased value estimator of $D_{\mathrm{KL}}[\pi_\theta(\cdot|s_t)\|\pi_{\mathrm{ref}}(\cdot|s_t)]$ when the sampled token is drawn from $\pi_\theta(\cdot|s_t)$.*

This proposition is about the scalar value of a sampled KL estimator. It does not by itself imply that placing $\hat{D}_t^{(k3)}$ as a local differentiable loss term gives the policy gradient of a KL-regularized autoregressive objective. Even in the on-policy case $\pi_{\theta_{\mathrm{old}}} = \pi_\theta$, a token a_i changes not only its own local KL term but also the distribution of future prefixes, and therefore the KL costs at later positions. The next section derives this missing causal term directly.

4 On-Policy f-Divergence Regularization

4.1 General Policy Gradient

Consider the on-policy regularized objective

$$Q_f(\theta) = \mathbb{E}_x\, \ell\big(\mathbb{E}_{a\sim\pi_\theta(\cdot|x)} r(x,a)\big) - \beta\, \mathbb{E}_x \mathbb{E}_{a\sim\pi_\theta(\cdot|x)} \sum_{i=1}^{T} f(u_i), \qquad u_i = \frac{\pi_{\mathrm{ref}}(a_i \mid s_i)}{\pi_\theta(a_i \mid s_i)}. \tag{5}$$

Here f is a differentiable convex generator applied to the token-level likelihood ratio; KL is only one special case. The function ℓ captures the prompt-level reward objective induced by the advantage construction. For binary-reward GRPO, Proposition 1 gives $\ell(p) = 2\arcsin\sqrt{p}$. Thus the derivation below applies to GRPO as a special case while also covering a broader class of token-wise f-divergence regularizers between the current and reference policies. The formulation should be read as a sampled autoregressive analogue of per-state f-divergence regularization: at each generated prefix s_i, the penalty is evaluated on the sampled action through the ratio $\pi_{\mathrm{ref}}(a_i|s_i)/\pi_\theta(a_i|s_i)$, and the expectation over trajectories supplies the state distribution induced by π_θ.

Theorem 4 (On-policy gradient with token-wise regularization)**.** *Assume differentiable policies with fixed common positive support, a fixed reference policy and reward, differentiable ℓ, and valid interchange of differentiation and expectation. For any action-independent baseline $b(s_i)$, the gradient of Eq.* (5) *is*

$$\nabla_\theta Q_f(\theta) = \mathbb{E}_x \mathbb{E}_{a\sim\pi_\theta(\cdot|x)} \sum_{i=1}^{T} w_i\, \nabla_\theta \log \pi_\theta(a_i \mid s_i), \qquad w_i = \mathrm{sg}(A_i - \beta\, \mathit{reg}_i)\,, \tag{6}$$

where

$$A_i = \ell'(\mathbb{E}_{a\sim\pi_\theta} r(x,a))\, r(x,a) - b(s_i), \tag{7}$$

$$\mathit{reg}_i = \sum_{j\geq i} f(u_j) - f'(u_i)u_i = \sum_{j>i} f(u_j) + f(u_i) - f'(u_i)u_i. \tag{8}$$

The proof is in Appendix C.3.1. Two features motivate FRPO. First, the reward signal carries the prompt-dependent gain $\ell'(\mathbb{E}\, r)$, while reg_i does not; the regularizer correction therefore belongs after advantage construction. Second, reg_i combines the direct derivative $-u_i f'(u_i)$ with the sampling-distribution contribution $\sum_{j\geq i} f(u_j)$. Its current term accounts for the action distribution, and its future terms account for later prefixes. A local differentiable loss omits this score-function contribution. In a one-step bandit the future terms disappear, but the current action-distribution term remains. These statements apply to general f, not only reverse KL.

4.2 Specialization to GRPO and Reverse KL

Corollary 5 (GRPO advantage as the score of the arcsin utility)**.** *Theorem 4 specializes to binary-reward GRPO via $\ell(p) = 2\arcsin\sqrt{p}$, with the population GRPO advantage $A_i^{\mathrm{GRPO}} = (r(x,a) - p_\theta(x))/\sqrt{p_\theta(x)(1-p_\theta(x))} = \ell'(p_\theta(x))(r(x,a) - p_\theta(x))$ recovered as the score of ℓ. Since reg_i is unmultiplied by ℓ', any reward-side modification is rescaled by $\ell'(p)$ with r while a token-level regularizer is not, so the only placement consistent with both signals is* after *advantage construction.*

The GRPO advantage and the future-KL correction are thus objects of *different scaling type* under ℓ and cannot be merged into a single reward without distorting one of them. Section 5 turns this into a one-line algorithmic change.

For token-wise reverse KL, taking $f(u) = -\log u$ gives, up to action-independent constants that vanish under the score-function identity,

$$\nabla_\theta Q_{\mathrm{rKL}}(\theta) = \mathbb{E}_x \mathbb{E}_{a\sim\pi_\theta} \sum_{i=1}^{T} \left[A_i^{\mathrm{GRPO}} - \beta \sum_{j\geq i} \log \frac{\pi_\theta(a_j \mid s_j)}{\pi_{\mathrm{ref}}(a_j \mid s_j)} \right] \nabla_\theta \log \pi_\theta(a_i \mid s_i). \tag{9}$$

Thus each token receives the outcome advantage plus a causal future-KL correction: early tokens are penalized for the regularization cost they induce downstream, not only for their own log-ratio.

The same theorem also covers the $k3$ generator $f(u) = -\log u + u - 1$. Substituting it into Eq. (8) yields the coefficient

$$A_i^{\mathrm{GRPO}} - \beta \left[-\log u_i + \sum_{j>i} \left(-\log u_j + u_j - 1 \right) \right]. \tag{10}$$

Unlike a local $k3$ loss term, which differentiates only the token penalty, Eq. (10) also includes the future regularization terms. This matters because $k3$ was designed primarily as a low-variance nonnegative value estimator for KL, whereas here the regularizer also acts as a token-level credit signal, so the sign and centering of the cumulative term affect how blame or credit is assigned to earlier tokens.

4.3 What the Loss-Side Surrogate Misses

The classical loss-side implementation in Eq. (2) treats $\hat{D}_{i,t}$ as an additive differentiable token loss. For example,

$$\nabla_\theta \hat{D}_t^{(k3)} = (1 - u_t)\, \nabla_\theta \log \pi_\theta(a_t \mid s_t).$$

This differentiates the penalty at a fixed sampled token and prefix, omitting the sampling-distribution contribution in Eq. (8), including the future cost-to-go. The omission persists even on-policy and is separate from rollout staleness.

Proposition 6 (Local loss-side KL omits sampling-distribution terms)**.** *For Eq.* (5)*, differentiating only the sampled local penalty $f(u_i)$ recovers $-f'(u_i)u_i$ but omits the score coefficient $\sum_{j\geq i} f(u_j)$. Its expected update therefore differs from the objective's gradient unless the omitted score contribution has zero expectation.*

For the reverse $k3$ estimator $f(u) = -\log u + u - 1$, the local differentiable loss gives the coefficient $1 - u_i$. The on-policy trajectory objective instead gives

$$\mathrm{reg}_i = -\log u_i + \sum_{j>i} \left(-\log u_j + u_j - 1 \right),$$

where the second term is the missing future regularization cost. For the forward $k3$ estimator $f(u) = u \log u - u + 1$, the same calculation gives

$$\mathrm{reg}_i = 1 - u_i + \sum_{j>i} (u_j \log u_j - u_j + 1),$$

again showing that the missing term is not an artifact of reverse KL. These expressions are the regularization analogues of reward-to-go in policy gradient: past costs can be dropped by the score-function identity, but future costs cannot.

Mini-batch updates from one rollout. Sequential mini-batches reuse the rollout-time advantage. Both reward and future-KL credit receive the same ratio $\rho_{i,t}$; clipping gates their combined weight. With population group statistics and the full horizon, the surrogate matches the regularized objective's gradient at θ_{old}. Under the smoothness assumptions of Appendix C.5, its local gradient error is $O(\|\theta - \theta_{\mathrm{old}}\|)$ while clipping is inactive; finite-batch sampling and normalization are separate effects. Empirically, 0.17% of token updates are gated, recomputing future KL changes pass@16 by -0.5 points, within seed variation, and four updates per rollout achieve similar accuracy to strictly on-policy training with matched loss aggregation (Appendix F.12).

5 Future-KL GRPO and Optimality View

Equation (9) suggests a lightweight modification of GRPO: compute the group-normalized outcome advantage from the original verifier reward, and then add a token-dependent future-KL correction. For a window spanning W response tokens including the current token, define

$$\begin{aligned} x_{i,W}(t) &:= \sum_{j=t}^{\min\{t+W-1,T_i\}} \delta_{i,j}, \qquad \delta_{i,j} := \log \frac{\pi_\theta(a_{i,j} \mid s_{i,j})}{\pi_{\mathrm{ref}}(a_{i,j} \mid s_{i,j})}, \\ \hat{A}_{i,t}^{FRPO(W)} &:= \mathrm{sg}\Big(\hat{A}_i^{\mathrm{GRPO}} - \beta x_{i,W}(t) \Big). \end{aligned} \tag{11}$$

Here $W = 1$ gives current-token credit placed inside the advantage rather than an additive loss term, a finite W gives truncated downstream credit, and $W = \infty$ denotes the full cost-to-go in Eq. (9); the whole-sequence alternative additionally includes past terms and assigns one response-level value to every token. Full-horizon FRPO, with $\hat{A}^{FRPO}_{i,t} := \hat{A}^{FRPO(\infty)}_{i,t}$, is the objective-derived default used in the main experiments. Finite windows omit some action-dependent future credit, but provide a practical approximation parameter controlling the magnitude, variability, and downstream information in the sampled scalar weight.

For efficiency, the default implementation evaluates $\delta_{i,j}$ once per rollout batch at $\pi_{\theta_{\text{old}}}$ and then optimizes the usual clipped PPO surrogate with $\hat{A}^{FRPO(W)}_{i,t}$; see Appendix D. Because the future-KL correction is part of the advantage, the PPO update uses $\rho_{i,t}\hat{A}^{FRPO(W)}_{i,t}$ (and its clipped counterpart), so both the verifier reward signal and the regularization signal receive the same importance-ratio correction from $\pi_{\theta_{\text{old}}}$ to π_θ. This decoupled form preserves GRPO's outcome-reward normalization while restoring token-level credit assignment from the regularizer.

Simple change, structural reason. Equation (11) adds about ten lines of code to a standard GRPO trainer: one reverse cumulative sum over per-token log-ratios whose log-probabilities the trainer already computes, with no extra forward passes and no learned value function. The non-trivial part is the placement: by Corollary 5, the only placement consistent with the on-policy gradient under non-linear ℓ is after advantage construction, whereas KL-in-reward re-weights the regularizer by $\ell'(p)$ across prompts (Proposition 2; the all-equal-group amplification in Section 3 explains why it collapses in practice), and loss-side KL omits the future cost-to-go even on-policy (Proposition 6). Theorem 15(b) gives local improvement when the corrected signal yields the target gradient; part (a) gives a separate fixed-reward bound under explicit action-probability movement conditions.

Appendix E.1 discusses alternative ways of inserting KL before GRPO normalization: trajectory-level KL-in-reward changes the group mean and standard deviation, while step-level normalization is fragile for small groups and variable-length responses.

Optimality view. The same future-KL structure also follows from the optimality conditions of entropy-regularized RL. For a fixed prompt, the maximizer of $\mathbb{E}_{a\sim\pi}[r(x,a)] - \beta\, D_{\text{KL}}[\pi\|\pi_{\text{ref}}]$ is $\pi^* \propto \pi_{\text{ref}} \exp(r/\beta)$, and at π^* the process reward after the prefix $a^{(t)}$ is $r_t = r(x,a) - \beta \sum_{j>t} \log\big(\pi^*(a_j \mid x, a^{(j-1)})/\pi_{\text{ref}}(a_j \mid x, a^{(j-1)})\big)$ (Proposition 20, Appendix C.6), so each token log-ratio is exactly the change $r_t - r_{t-1}$ in process reward. This view is not needed for Theorem 4 but gives the same interpretation: KL regularization is a process-level signal whose token weight is a future cost-to-go, and FRPO approximates this structure on-policy with sampled log-ratios and GRPO's advantage in place of a learned process value.

6 Experiments and Results

Basic Configuration We select Qwen3-1.7B-Base (Yang et al., 2025) as the dense model and Qwen3-30B-A3B-Base as the MoE model. For the training dataset, we use a filtered version of DAPO-Math-17k (Yu et al., 2025), which contains about 12,000 problems. For the evaluation benchmarks, we choose MATH500 (Hendrycks et al., 2021), AIME24, AIME25, AMC23, OlympiadBench (He et al., 2024), and MinervaMath (Lewkowycz et al., 2022). For pass@16, we sample with replacement 1,000 times and report the evaluation-bootstrap mean and standard deviation; mean@16 is the average accuracy over the 16 samples per problem. Hyperparameters and further results are in Appendix F.

6.1 Main Experiments

Figure 2 shows that future-KL credit improves pass@n for Qwen3-30B-A3B-Base by nearly five percentage points over the strongest baseline, with higher final entropy and lower PPO-KL. Around step 70, its responses grow from about 1,000 to 3,000 tokens together with a jump in both pass@16 and mean@16 (Figure 12), whereas the baselines stay near 1,000 tokens or shrink, and KL-in-reward collapses to near-empty responses. Table 1 sharpens this pattern: at 30B, loss-side KL is below no-KL on all six pass@16 benchmarks, whereas full-horizon FRPO is above both on all six. It also

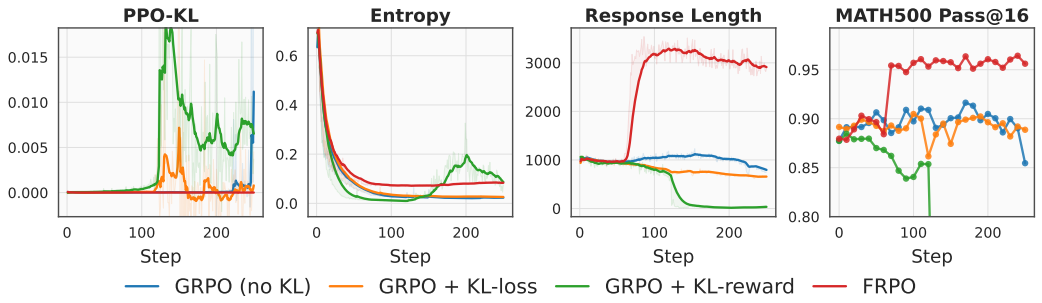


Figure 2: Training dynamics of Qwen3-30B-A3B-Base with different KL integrations, each at its default coefficient (Table 1). PPO-KL is the trainer's token-averaged KL between the actor and the rollout policy, not KL to the reference policy.

exceeds loss-side KL on all six mean@16 benchmarks (Table 13), so the pass@16 gain is not solely a higher-diversity effect.

Choice of coefficients. Each placement is run at a fixed coefficient that is not re-tuned per model: $\beta = 10^{-2}$ for FRPO, selected in a preliminary sweep; 10^{-3} for the loss-side penalty, the default of verl; and 10^{-5} for KL-in-reward (Appendix F.1). The 1.7B sweep of Appendix F.8 confirms $\beta = 10^{-2}$ as the best of the swept values for FRPO on both MATH500 metrics. At every matched β from 10^{-4} to 10^{-1}, FRPO matches the accuracy of the loss-side penalty within run-to-run spread, and exceeds it under the strongest regularization (48.7% versus 47.1% mean@16 at $\beta = 10^{-1}$), while keeping the policy 8–27% closer to the reference and generating no longer responses; the future-KL credit therefore does not inflate response length. KL-in-reward, in contrast, fails at every coefficient we tried (Appendix F.6): once KL enters the normalized reward, lowering β cannot remove its $O(1)$ effect (Remark 13).

In the Qwen3-1.7B-Base horizon sweep of Appendix F.7, compared at step 600, every FRPO horizon exceeds both GRPO baselines on MATH500: the full horizon reaches 68.1% mean@16 and 92.0% pass@16, and the window $W = 512$ the best 70.3% and 93.2%, versus 67.4% and 86.0% without KL and 66.4% and 85.4% with the loss-side penalty. Every token-level variant also exceeds whole-sequence KL credit by 6.9–9.1 points in MATH500 mean@16, supporting token-level placement of the regularization credit.

Table 1: Pass@16 (%) of Qwen3-1.7B/30B-A3B-Base: evaluation-bootstrap mean (standard deviation in parentheses). All FRPO rows use the full horizon ($W = \infty$). Each method uses its default coefficient (FRPO 10^{-2}, KL-loss 10^{-3}, KL-reward 10^{-5}).

Model	MATH500	AIME24	AIME25	AMC23	OlympiadBench	MinervaMath
Qwen3-1.7B-Base	69.65 (2.48)	12.94 (2.87)	7.15 (2.45)	52.76 (4.01)	38.80 (0.75)	36.02 (1.16)
+ GRPO (no KL)	78.62 (2.01)	25.82 (2.92)	16.76 (2.72)	69.73 (2.97)	57.23 (0.58)	**48.60** (0.83)
+ GRPO KL-loss	77.07 (1.68)	**27.15** (2.65)	18.33 (3.26)	**81.56** (3.23)	54.90 (0.56)	46.36 (0.90)
+ GRPO KL-reward	collapse	/	/	/	/	/
+ FRPO	**85.30** (3.00)	27.00 (2.53)	**20.31** (3.73)	74.47 (3.00)	**59.13** (0.65)	47.15 (1.03)
Qwen3-30B-A3B-Base	87.77 (2.08)	35.31 (3.38)	21.04 (2.39)	76.67 (2.44)	61.36 (0.76)	51.68 (0.95)
+ GRPO (no KL)	91.63 (0.87)	53.06 (2.99)	27.45 (2.30)	91.44 (1.31)	66.33 (0.49)	57.46 (0.68)
+ GRPO KL-loss	90.23 (1.17)	51.25 (4.52)	22.04 (3.05)	89.06 (2.57)	62.86 (0.49)	55.51 (0.70)
+ GRPO KL-reward	collapse	/	/	/	/	/
+ FRPO	**96.07** (1.54)	**63.88** (3.86)	**45.23** (3.41)	**94.25** (1.35)	**77.71** (0.47)	**58.65** (0.71)

Ablations. Appendix F.3 compares KL directions and $k1$ versus $k3$ future-KL estimators; reverse KL with signed $k1$ is the FRPO default because the cumulative term acts as credit rather than only a nonnegative divergence monitor. Figure 4 gives a complementary 30B KL-in-reward comparison using GAE, and Appendix F.7 compares truncated, full-horizon, and whole-sequence credit. Appendix F.6 relates the KL-reward collapse to a finite-group effect: most groups have identical verifier rewards yet receive an $O(1)$, β-independent KL-driven advantage.

7 Related Work

RL for LLM Policy Optimization In the classical RL domain, TRPO (Schulman et al., 2015) and PPO (Schulman et al., 2017) are generally utilized to optimize models. As LLMs become popular, RL algorithms are applied to further improve their abilities (Rafailov et al., 2023; Jaech et al., 2024; Ouyang et al., 2022; Bai et al., 2022). Since the release of DeepSeek-R1 (Guo et al., 2025), critic-free policy optimization (Liu et al., 2025; Hu et al., 2025) has become more and more popular in optimizing LLM, as it could be formulated into a bandit problem in a vast number of scenarios, leading to simple adaptations to many training settings. Based on the initially proposed GRPO (Shao et al., 2024; Guo et al., 2025) algorithm, many follow-up variants appear to resolve the bias (Liu et al., 2025; Yang et al., 2026), mitigate the variance (Qi et al., 2026; Luo et al., 2026), improve the sample efficiency (Yao et al., 2025; Xiong et al., 2025b), and derive simpler and more effective algorithms (Xiong et al., 2025a; Yu et al., 2025; Zhang et al., 2025a; Liu et al., 2026; Zheng et al., 2025).

Where the KL Term Enters In RLHF pipelines built on PPO or REINFORCE, the KL penalty is typically folded into the per-token reward and propagated to earlier tokens through the return, either by a learned value function with GAE (Ouyang et al., 2022) or by Monte-Carlo reward-to-go as in RLOO and REINFORCE++ (Ahmadian et al., 2024; Hu et al., 2025). In the linear-reward objective these pipelines target, this reward-shaping route already supplies future-KL credit, and Zhang et al. (2025b) analyze the resulting estimators and their off-policy corrections. GRPO instead introduced KL as a separate loss-side token penalty (Shao et al., 2024), which is local by construction. Our contribution is not the observation that regularization has a cost-to-go, but that under GRPO's group normalization the reward-shaping route and the loss-side route are both inconsistent with the on-policy gradient of the induced non-linear objective, and that the consistent placement is after advantage construction (Section 4).

Entropy-Regularized RL Theoretically, Zhao et al. (2024) establish a sharp convergence rate compared with unregularized RL. Other works (Huang et al., 2024; Xiong et al., 2023; Vieillard et al., 2020a; Liu et al., 2019; Wen et al., 2024; Eysenbach et al., 2018; Della Vecchia et al., 2022; Vieillard et al., 2020b) also show the effectiveness of regularization analytically and empirically. Entropy regularization is also used beyond LLMs (Tiapkin et al., 2024; Kozuno et al., 2022), as it limits deviation from a strong base model and preserves out-of-domain generalization.

LLM for Reasoning RL Due to the strong capabilities of pretrained LLMs, they have been used to tackle challenging scenarios requiring complex reasoning (Wei et al., 2022a; Jin et al., 2025; Kumar et al., 2024). Among various reasoning domains, mathematical reasoning is particularly important due to its ease of verification and its foundational role for other tasks (Wei et al., 2022b; Shao et al., 2024; Yang et al., 2024; Ahn et al., 2024; Lightman et al., 2023; Yuan et al., 2024; Cui et al., 2025). The base models, evaluation metrics, and other supporting techniques are relatively mature; therefore, we also select math reasoning as the empirical task to verify the effectiveness of our proposed method. Commonly used benchmarks (Hendrycks et al., 2021; He et al., 2024; Lewkowycz et al., 2022) and open-sourced datasets (Yu et al., 2025; LI et al., 2024) are also available for conducting large-scale experiments.

8 Conclusion and Discussion

We revisited KL regularization in GRPO from the on-policy objective rather than only from the implementation of a local loss penalty. The resulting policy-gradient formula applies to general token-wise f-divergence regularizers and shows that autoregressive regularization induces a future cost-to-go. For reverse KL, this gives the future-KL correction used by FRPO, which keeps GRPO's group-normalized outcome advantage separate from token-level regularization credit. The same structure is supported by the optimality view of KL-regularized trajectory rewards. In our main large-model setting, this formulation improves pass@n and mean@n over loss-side KL while maintaining higher entropy and smaller PPO updates. At matched coefficients, FRPO reaches the accuracy of loss-side KL with a policy closer to the reference, whereas KL-in-reward fails at every coefficient. The horizon further provides a practical control: an intermediate window gives the best late-training accuracy of all runs in our 1.7B sweep.

AI USE STATEMENT

In this work, we used generative AI tools for polishing the paper writing and searching for relevant references. We also used LLM-based assistants to help check the proofs of supporting results in the appendix (e.g., Appendix C.5), and to implement analysis code and diagnostic experiments. We have not used generative AI tools for generating references, method proposals, etc. We have reviewed all AI-assisted work. We checked the retrieved references by manually searching for the correct information, and all AI-assisted proofs, code, and reported numbers were checked by the authors against the derivations, training logs, and evaluation outputs. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

All authors have read and adhere to the ICLR Code of Ethics. This work studies the optimization objective of KL-regularized reinforcement learning for LLM post-training and proposes a modified advantage computation for GRPO. It does not involve human subjects, crowdsourcing, or user studies, and we collect no new data. Training uses a filtered subset of the publicly available DAPO-Math-17k dataset, and evaluation uses public mathematical benchmarks (MATH500, AIME24, AIME25, AMC23, OlympiadBench, and MinervaMath). These datasets consist of mathematics problems with reference answers and, to our knowledge, contain no personally identifiable or sensitive information. All base models are publicly released open-weight Qwen3 series models, and all assets are used in accordance with their licenses.

REPRODUCIBILITY STATEMENT

We have made the following efforts to ensure reproducibility. **Theory:** The problem setup and notation are given in Section 2, and the assumptions of each theoretical result (e.g., binary verifier rewards, population group statistics, and on-policy sampling) are stated in its formal statement in Sections 3–5. Complete proofs are provided in Appendix B (arcsin utility of binary GRPO) and Appendix C (KL-in-reward distortion, the on-policy regularized policy gradient in Theorem 4, conditional policy improvement with PPO clipping, sequential mini-batch updates from one rollout, and the optimality view), and the expectation and gradient properties of the KL estimators are derived in Appendix A. **Algorithm:** The FRPO update is defined in Eq. (11), and full pseudocode together with implementation notes on token masking, windowing, and estimator choice is given in Algorithm 1 (Appendix D). **Experiments:** Section 6 specifies the base models, training data, evaluation benchmarks, and the bootstrap protocol for pass@16. Appendix F reports the hyperparameters for the baselines and FRPO (Tables 5 and 6), the prompt template, benchmark sizes (Table 12), the hardware and inference backend, the protocol of the horizon sweep (Appendix F.7), how the regularization coefficients were chosen (Appendix F.1), multi-run results at matched training length (Appendix F.2), the all-equal-group diagnostic of KL-in-reward (Appendix F.6), the sensitivity to the regularization coefficient (Appendix F.8), and the mini-batch staleness and clipping diagnostics (Appendix F.12). All models and datasets used are publicly available. **Code:** We plan to open-source our code, built on the verl framework, including the FRPO implementation, training and evaluation scripts with the configurations used in our experiments, and the filtered training data.

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.

Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. Large language models for mathematical reasoning: Progresses and challenges. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pp. 225–237, 2024.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless

assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

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

Riccardo Della Vecchia, Alena Shilova, Philippe Preux, and Riad Akrour. Entropy regularized reinforcement learning with cascading networks. *arXiv preprint arXiv:2210.08503*, 2022.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

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, 2025. URL `https://arxiv.org/abs/2505.24298`.

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

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 2024.

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

Jian Hu, Jason Klein Liu, Haotian Xu, and Wei Shen. Reinforce++: Stabilizing critic-free policy optimization with global advantage normalization. *arXiv preprint arXiv:2501.03262*, 2025.

Audrey Huang, Wenhao Zhan, Tengyang Xie, Jason D Lee, Wen Sun, Akshay Krishnamurthy, and Dylan J Foster. Correcting the mythos of kl-regularization: Direct alignment without overoptimization via chi-squared preference optimization. *arXiv preprint arXiv:2407.13399*, 2024.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

Tadashi Kozuno, Wenhao Yang, Nino Vieillard, Toshinori Kitamura, Yunhao Tang, Jincheng Mei, Pierre Ménard, Mohammad Gheshlaghi Azar, Michal Valko, Rémi Munos, et al. Kl-entropy-regularized rl with a generative model is minimax optimal. *arXiv preprint arXiv:2205.14211*, 2022.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, et al. Training language models to self-correct via reinforcement learning. *arXiv preprint arXiv:2409.12917*, 2024.

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 ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

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

Jia LI, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. Numinamath. `https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf`, 2024.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The twelfth international conference on learning representations*, 2023.

Jingbin Liu, Xinyang Gu, and Shuai Liu. Policy optimization reinforcement learning with entropy regularization. *arXiv preprint arXiv:1912.01557*, 2019.

Shih-Yang Liu, Xin Dong, Ximing Lu, Shizhe Diao, Peter Belcak, Mingjie Liu, Min-Hung Chen, Hongxu Yin, Yu-Chiang Frank Wang, Kwang-Ting Cheng, et al. Gdpo: Group reward-decoupled normalization policy optimization for multi-reward rl optimization. *arXiv preprint arXiv:2601.05242*, 2026.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

Yu Luo, Shuo Han, Yihan Hu, Dong Li, and Jianye Hao. Ratio-variance regularized policy optimization for efficient llm fine-tuning. *arXiv preprint arXiv:2601.03320*, 2026.

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

Penghui Qi, Xiangxin Zhou, Zichen Liu, Tianyu Pang, Chao Du, Min Lin, and Wee Sun Lee. Rethinking the trust region in llm reinforcement learning. *arXiv preprint arXiv:2602.04879*, 2026.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

John Schulman. Approximating kl divergence. `http://joschu.net/blog/kl-approx.html`, 2020. Blog post.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015.

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

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

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

Daniil Tiapkin, Nikita Morozov, Alexey Naumov, and Dmitry P Vetrov. Generative flow networks as entropy-regularized rl. In *International Conference on Artificial Intelligence and Statistics*, pp. 4213–4221. PMLR, 2024.

Nino Vieillard, Tadashi Kozuno, Bruno Scherrer, Olivier Pietquin, Rémi Munos, and Matthieu Geist. Leverage the average: an analysis of kl regularization in reinforcement learning. *Advances in Neural Information Processing Systems*, 33:12163–12174, 2020a.

Nino Vieillard, Bruno Scherrer, Olivier Pietquin, and Matthieu Geist. Momentum in reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 2529–2538. PMLR, 2020b.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022a.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022b.

Muning Wen, Junwei Liao, Cheng Deng, Jun Wang, Weinan Zhang, and Ying Wen. Entropy-regularized token-level policy optimization for language agent reinforcement. *arXiv preprint arXiv:2402.06700*, 2024.

Wei Xiong, Hanze Dong, Chenlu Ye, Ziqi Wang, Han Zhong, Heng Ji, Nan Jiang, and Tong Zhang. Iterative preference learning from human feedback: Bridging theory and practice for rlhf under kl-constraint. *arXiv preprint arXiv:2312.11456*, 2023.

Wei Xiong, Jiarui Yao, Yuhui Xu, Bo Pang, Lei Wang, Doyen Sahoo, Junnan Li, Nan Jiang, Tong Zhang, Caiming Xiong, et al. A minimalist approach to llm reasoning: from rejection sampling to reinforce. *arXiv preprint arXiv:2504.11343*, 2025a.

Wei Xiong, Chenlu Ye, Baohao Liao, Hanze Dong, Xinxing Xu, Christof Monz, Jiang Bian, Nan Jiang, and Tong Zhang. Reinforce-ada: An adaptive sampling framework under non-linear rl objectives. *arXiv preprint arXiv:2510.04996*, 2025b.

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

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, 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.

Fengkai Yang, Zherui Chen, Xiaohan Wang, Xiaodong Lu, Jiajun Chai, Guojun Yin, Wei Lin, Shuai Ma, Fuzhen Zhuang, Deqing Wang, et al. Your group-relative advantage is biased. *arXiv preprint arXiv:2601.08521*, 2026.

Jiarui Yao, Yifan Hao, Hanning Zhang, Hanze Dong, Wei Xiong, Nan Jiang, and Tong Zhang. Optimizing chain-of-thought reasoners via gradient variance minimization in rejection sampling and rl. *arXiv preprint arXiv:2505.02391*, 2025.

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

Lifan Yuan, Wendi Li, Huayu Chen, Ganqu Cui, Ning Ding, Kaiyan Zhang, Bowen Zhou, Zhiyuan Liu, and Hao Peng. Free process rewards without process labels. *arXiv preprint arXiv:2412.01981*, 2024.

Enci Zhang, Xingang Yan, Wei Lin, Tianxiang Zhang, and Lu Qianchun. Learning like humans: Advancing llm reasoning capabilities via adaptive difficulty curriculum learning and expert-guided self-reformulation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 6630–6644, 2025a.

Yifan Zhang, Yifeng Liu, Huizhuo Yuan, Yang Yuan, Quanquan Gu, and Andrew Chi-Chih Yao. On the design of kl-regularized policy gradient algorithms for llm reasoning. *arXiv preprint arXiv:2505.17508*, 2025b.

Heyang Zhao, Chenlu Ye, Quanquan Gu, and Tong Zhang. Sharp analysis for kl-regularized contextual bandits and rlhf. *arXiv preprint arXiv:2411.04625*, 2024.

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

A KL Estimators

A.1 Reverse KL Estimator Variants and Their Properties

Following Schulman (2020), we analyze four estimators. Since the original GRPO objective and popular RL frameworks like verl use reverse KL as the default option, we first summarize the expectations and gradients for reverse KL-based estimators, i.e., $D_{\mathrm{KL}}(\pi_\theta \| \pi_{\mathrm{ref}})$.

Definition 7 (KL Estimators)**.**

$$\begin{aligned}
\hat{D}^{(k1)} &= \delta = \log \alpha, \\
\hat{D}^{(k2)} &= \tfrac{1}{2}\delta^2 = \tfrac{1}{2}(\log \alpha)^2, \\
\hat{D}^{(k3)} &= \alpha^{-1} + \delta - 1 = \alpha^{-1} + \log \alpha - 1, \\
\hat{D}^{(\text{full})} &= \sum_{a \in \mathcal{V}} \pi_\theta(a) \log \frac{\pi_\theta(a)}{\pi_{\mathrm{ref}}(a)}.
\end{aligned}$$

$k1$ is used in aligning InstructGPT (Ouyang et al., 2022), while $k3$ is used in DeepSeek-Math (Shao et al., 2024).

A.1.1 Expectation Properties

Under $\mathbb{E}_{\pi_\theta}$:

- $\mathbb{E}_{\pi_\theta}[\hat{D}^{(k1)}] = D_{\mathrm{KL}}[\pi_\theta \| \pi_{\mathrm{ref}}]$. **Unbiased**, but can be negative for individual tokens.
- $\mathbb{E}_{\pi_\theta}[\hat{D}^{(k2)}] = \frac{1}{2}\mathrm{Var}_{\pi_\theta}[\delta] + \frac{1}{2}(D_{\mathrm{KL}})^2 \neq D_{\mathrm{KL}}$ in general. **Biased** as a value estimator.
- $\mathbb{E}_{\pi_\theta}[\hat{D}^{(k3)}] = D_{\mathrm{KL}}[\pi_\theta \| \pi_{\mathrm{ref}}]$ (Proposition 3). **Unbiased** and always ≥ 0.
- $\hat{D}^{(\text{full})}$ is **exact** (no sampling involved).

A.1.2 Gradient Properties

The gradient of the expected KL loss under the sampling distribution $\pi_{\theta_{\mathrm{old}}}$ depends critically on the estimator choice. We analyze $\nabla_\theta \mathbb{E}_{\pi_{\theta_{\mathrm{old}}}}[\hat{D}]$ for each estimator, noting that π_{ref} is constant w.r.t. θ.

k1 gradient. Since $\nabla_\theta \hat{D}^{(k1)} = \nabla_\theta \log \pi_\theta(o_t \mid s_t)$, the expected gradient under π_θ is:

$$\mathbb{E}_{\pi_\theta}\left[\nabla_\theta \hat{D}^{(k1)}\right] = \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta] = \sum_{o \in \mathcal{V}} \pi_\theta(o|s) \frac{\nabla_\theta \pi_\theta(o|s)}{\pi_\theta(o|s)} = \nabla_\theta \sum_{o \in \mathcal{V}} \pi_\theta(o|s) = 0.$$

This is the score function identity—the expected gradient of $k1$ is zero under π_θ, *not* $\nabla_\theta D_{\mathrm{KL}}$. Under $\pi_{\theta_{old}}$ (without importance weighting), $\mathbb{E}_{\pi_{\theta_{\mathrm{old}}}}[\nabla_\theta \hat{D}^{(k1)}]$ gives a **biased** gradient estimate of $\nabla_\theta D_{\mathrm{KL}}[\pi_\theta \| \pi_{\mathrm{ref}}]$.

k2 gradient. We have $\nabla_\theta \hat{D}^{(k2)} = \delta \cdot \nabla_\theta \delta = \delta \cdot \nabla_\theta \log \pi_\theta$.

Proposition 8. *Under on-policy sampling, $k2$ provides an unbiased gradient estimator:* $\mathbb{E}_{\pi_\theta}\left[\nabla_\theta \hat{D}^{(k2)}\right] = \nabla_\theta D_{\mathrm{KL}}[\pi_\theta \| \pi_{\mathrm{ref}}]$.

Proof. Since $\nabla_\theta \delta = \nabla_\theta \log \pi_\theta$ (the π_{ref} term vanishes):

$$\begin{aligned}
\nabla_\theta D_{\mathrm{KL}}[\pi_\theta \| \pi_{\mathrm{ref}}] &= \nabla_\theta \sum_a \pi_\theta(a) \log \frac{\pi_\theta(a)}{\pi_{\mathrm{ref}}(a)} \\
&= \sum_a \nabla_\theta \pi_\theta(a) \cdot \log \frac{\pi_\theta(a)}{\pi_{\mathrm{ref}}(a)} + \sum_a \pi_\theta(a) \cdot \frac{\nabla_\theta \pi_\theta(a)}{\pi_\theta(a)} \\
&= \sum_a \pi_\theta(a) \cdot \nabla_\theta \log \pi_\theta(a) \cdot \delta_a + \underbrace{\sum_a \nabla_\theta \pi_\theta(a)}_{=0} \\
&= \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta \cdot \delta] = \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \hat{D}^{(k2)}\right].
\end{aligned}$$

□

Moreover, under $\pi_{\theta_{\mathrm{old}}} \approx \pi_\theta$ (e.g., early stage in PPO epochs), it remains approximately unbiased. Under off-policy sampling ($\pi_{\theta_{\mathrm{old}}} \neq \pi_\theta$), the $k2$ gradient estimator also becomes biased, though the bias is generally smaller than that of $k1$ or $k3$.

k3 gradient. We have:

$$\nabla_\theta \hat{D}^{(k3)} = \nabla_\theta \left[\frac{1}{\alpha} + \log \alpha - 1\right] = \left(1 - \frac{1}{\alpha}\right) \nabla_\theta \log \pi_\theta.$$

The expected gradient under π_θ:

$$\begin{aligned}
\mathbb{E}_{\pi_\theta}\left[\nabla_\theta \hat{D}^{(k3)}\right] &= \mathbb{E}_{\pi_\theta}\left[\left(1 - \frac{\pi_{\mathrm{ref}}}{\pi_\theta}\right) \nabla_\theta \log \pi_\theta\right] \\
&= \sum_a \nabla_\theta \pi_\theta(a) - \sum_a \pi_{\mathrm{ref}}(a) \frac{\nabla_\theta \pi_\theta(a)}{\pi_\theta(a)} \\
&= 0 - \mathbb{E}_{\pi_{\mathrm{ref}}}\left[\frac{\nabla_\theta \pi_\theta}{\pi_\theta}\right] \\
&= -\mathbb{E}_{\pi_{\mathrm{ref}}}\left[\frac{\nabla_\theta \pi_\theta}{\pi_\theta}\right].
\end{aligned}$$

In general, this does *not* equal $\nabla_\theta D_{\mathrm{KL}}[\pi_\theta \| \pi_{\mathrm{ref}}]$. Thus the expected gradient of $k3$ is **biased**.

Straight-through trick (verl's $k3+$ (Sheng et al., 2024)). Define:

$$\hat{D}^{(k3+)} = \underbrace{\hat{D}^{(k2)}}_{\text{backward}} - \mathrm{sg}\left(\hat{D}^{(k2)}\right) + \mathrm{sg}\left(\hat{D}^{(k3)}\right),$$

where $\mathrm{sg}(\cdot)$ denotes stop-gradient. This yields:

$$\begin{aligned}
\text{Forward value:} \quad & \hat{D}^{(k3+)} = \hat{D}^{(k3)}, \\
\text{Backward gradient:} \quad & \nabla_\theta \hat{D}^{(k3+)} = \nabla_\theta \hat{D}^{(k2)} = \delta \cdot \nabla_\theta \log \pi_\theta.
\end{aligned}$$

This combines the unbiased *value* of $k3$ with the unbiased *gradient* of $k2$.

We briefly summarize the properties of different KL estimators in Table 2.

A.2 Forward KL Estimators

For the forward KL formulation $\mathbb{D}_{\mathrm{KL}}(\pi_{\mathrm{ref}} \| \pi_\theta)$, we can similarly define KL estimators by reversing the log-ratio:

Table 2: Properties of Reverse KL estimators under $\mathbb{E}_{\pi_\theta}$. All sample-based estimators become biased under $\mathbb{E}_{\pi_{\theta_{\text{old}}}}$ when $\pi_{\theta_{\text{old}}} \neq \pi_\theta$.

Estimator	Value unbiased	Gradient unbiased	Non-negative
$k1$	✓	×	×
$k2$	×	✓	✓
$k3$	✓	×	✓
$k3+$	✓	✓	✓
full	exact	exact	✓

Definition 9 (Forward KL Estimators)**.**

$$\hat{D}^{(k1)}_{\text{forward}} = \log \alpha^{-1} = \log u = -\delta,$$
$$\hat{D}^{(k2)}_{\text{forward}} = \frac{1}{2}(-\delta)^2 = \frac{1}{2}\delta^2 = \frac{1}{2}(\log \alpha)^2,$$
$$\hat{D}^{(k3)}_{\text{forward}} = \alpha - \delta - 1,$$
$$\hat{D}^{(\text{full})}_{\text{forward}} = \sum_{a \in \mathcal{V}} \pi_{\text{ref}}(a) \log \frac{\pi_{\text{ref}}(a)}{\pi_\theta(a)}.$$

A.2.1 Expectation properties.

Under $\mathbb{E}_{\pi_{\text{ref}}}$:

$$\mathbb{E}_{\pi_{\text{ref}}}\left[\hat{D}^{(k1)}_{\text{forward}}\right] = \mathbb{E}_{\pi_{\text{ref}}}[-\delta] = \mathbb{D}_{\text{KL}}[\pi_{\text{ref}} \| \pi_\theta].$$

Thus $k1$ is an unbiased value estimator for the forward KL, but it can be negative for individual tokens.

For $k2$, we have

$$\mathbb{E}_{\pi_{\text{ref}}}\left[\hat{D}^{(k2)}_{\text{forward}}\right] = \frac{1}{2}\mathbb{E}_{\pi_{\text{ref}}}[\delta^2] = \frac{1}{2}\operatorname{Var}_{\pi_{\text{ref}}}[\delta] + \frac{1}{2}\left(\mathbb{D}_{\text{KL}}[\pi_{\text{ref}} \| \pi_\theta]\right)^2 \neq \mathbb{D}_{\text{KL}}[\pi_{\text{ref}} \| \pi_\theta]$$

in general. Hence $k2$ is biased as a value estimator.

For $k3$, since $\mathbb{E}_{\pi_{\text{ref}}}[\alpha] = 1$, we obtain

$$\mathbb{E}_{\pi_{\text{ref}}}\left[\hat{D}^{(k3)}_{\text{forward}}\right] = \mathbb{E}_{\pi_{\text{ref}}}[\alpha - \delta - 1] = \mathbb{E}_{\pi_{\text{ref}}}[-\delta] = \mathbb{D}_{\text{KL}}[\pi_{\text{ref}} \| \pi_\theta].$$

Moreover, by the inequality $x - \log x - 1 \geq 0$ for $x > 0$, $\hat{D}^{(k3)}_{\text{forward}} \geq 0$ for every token. Therefore, $k3$ is an unbiased and nonnegative value estimator for the forward KL.

Finally, $\hat{D}^{(full)}_{\text{forward}}$ is exact, since it explicitly sums over the whole vocabulary.

A.2.2 Gradient properties.

We now analyze

$$\nabla_\theta \mathbb{E}_{\pi_{\text{ref}}}\left[\hat{D}_{\text{forward}}\right],$$

noting that π_{ref} is constant with respect to θ.

k1 gradient. Since

$$\nabla_\theta \hat{D}^{(k1)}_{\text{forward}} = \nabla_\theta(-\delta) = -\nabla_\theta \log \pi_\theta,$$

we have

$$\mathbb{E}_{\pi_{\text{ref}}}\left[\nabla_\theta \hat{D}^{(k1)}_{\text{forward}}\right] = -\mathbb{E}_{\pi_{\text{ref}}}\left[\nabla_\theta \log \pi_\theta\right].$$

On the other hand,

$$\begin{aligned}\nabla_\theta \mathbb{D}_{\mathrm{KL}}[\pi_{\mathrm{ref}}\|\pi_\theta] &= \nabla_\theta \sum_a \pi_{\mathrm{ref}}(a) \log \frac{\pi_{\mathrm{ref}}(a)}{\pi_\theta(a)} \\ &= -\sum_a \pi_{\mathrm{ref}}(a) \nabla_\theta \log \pi_\theta(a) \\ &= -\mathbb{E}_{\pi_{\mathrm{ref}}}\left[\nabla_\theta \log \pi_\theta\right].\end{aligned}$$

Therefore,

$$\mathbb{E}_{\pi_{\mathrm{ref}}}\left[\nabla_\theta \hat{D}^{(k1)}_{\mathrm{forward}}\right] = \nabla_\theta \mathbb{D}_{\mathrm{KL}}[\pi_{\mathrm{ref}}\|\pi_\theta].$$

Thus $k1$ provides an unbiased gradient estimator for the forward KL under π_{ref} sampling.

k2 gradient. For $k2$, we have

$$\nabla_\theta \hat{D}^{(k2)}_{\mathrm{forward}} = \nabla_\theta \frac{1}{2}\delta^2 = \delta \cdot \nabla_\theta \log \pi_\theta.$$

Hence

$$\mathbb{E}_{\pi_{\mathrm{ref}}}\left[\nabla_\theta \hat{D}^{(k2)}_{\mathrm{forward}}\right] = \mathbb{E}_{\pi_{\mathrm{ref}}}\left[\delta \cdot \nabla_\theta \log \pi_\theta\right].$$

This is not equal to

$$\nabla_\theta \mathbb{D}_{\mathrm{KL}}[\pi_{\mathrm{ref}}\|\pi_\theta] = -\mathbb{E}_{\pi_{\mathrm{ref}}}\left[\nabla_\theta \log \pi_\theta\right]$$

in general. Therefore, although $k2$ is a second-order approximation to the KL value near $\alpha = 1$, it is generally biased as a gradient estimator for the forward KL.

k3 gradient. For $k3$, we have

$$\begin{aligned}\nabla_\theta \hat{D}^{(k3)}_{\mathrm{forward}} &= \nabla_\theta(\alpha - \delta - 1) \\ &= \alpha \nabla_\theta \log \pi_\theta - \nabla_\theta \log \pi_\theta \\ &= (\alpha - 1)\nabla_\theta \log \pi_\theta.\end{aligned}$$

Taking expectation under π_{ref} gives

$$\begin{aligned}\mathbb{E}_{\pi_{\mathrm{ref}}}\left[\nabla_\theta \hat{D}^{(k3)}_{\mathrm{forward}}\right] &= \mathbb{E}_{\pi_{\mathrm{ref}}}\left[(\alpha - 1)\nabla_\theta \log \pi_\theta\right] \\ &= \mathbb{E}_{\pi_{\mathrm{ref}}}\left[\alpha \nabla_\theta \log \pi_\theta\right] - \mathbb{E}_{\pi_{\mathrm{ref}}}\left[\nabla_\theta \log \pi_\theta\right] \\ &= \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta\right] - \mathbb{E}_{\pi_{\mathrm{ref}}}\left[\nabla_\theta \log \pi_\theta\right] \\ &= 0 - \mathbb{E}_{\pi_{\mathrm{ref}}}\left[\nabla_\theta \log \pi_\theta\right] \\ &= \nabla_\theta \mathbb{D}_{\mathrm{KL}}[\pi_{\mathrm{ref}}\|\pi_\theta].\end{aligned}$$

Thus $k3$ also provides an unbiased gradient estimator for the forward KL under π_{ref} sampling, while additionally being nonnegative and value-unbiased.

Under π_{ref} sampling, the forward KL estimators have the following properties in Table 3.

Table 3: Properties for Forward KL Estimators

Estimator	Value Unbiased	Gradient Unbiased	Nonnegative
$k1$	✓	✓	×
$k2$	×	×	✓
$k3$	✓	✓	✓
full	exact	exact	✓

B IMPLICIT ARCSIN OBJECTIVE OF BINARY GRPO

This appendix proves Proposition 1. We use the same notation as the main text. Fix a prompt x, let $a \sim \pi_\theta(\cdot|x)$ be a sampled response, and assume a binary verifier reward $r(x, a) \in \{0, 1\}$. Define

$$p_\theta(x) := \mathbb{E}_{a\sim\pi_\theta(\cdot|x)}[r(x, a)].$$

We analyze the population version of GRPO, replacing the empirical group mean and standard deviation by

$$\mu(x) = p_\theta(x), \qquad \sigma(x) = \sqrt{p_\theta(x)(1 - p_\theta(x))},$$

for $p_\theta(x) \in (0, 1)$. The corresponding population GRPO advantage is

$$A^{\mathrm{GRPO}}(x, a) = \frac{r(x, a) - p_\theta(x)}{\sqrt{p_\theta(x)(1 - p_\theta(x))}}.$$

Proof of Proposition 1. By the likelihood-ratio identity,

$$\nabla_\theta p_\theta(x) = \nabla_\theta \mathbb{E}_{a\sim\pi_\theta(\cdot|x)}[r(x, a)] = \mathbb{E}_{a\sim\pi_\theta(\cdot|x)}\left[r(x, a)\nabla_\theta \log \pi_\theta(a|x)\right].$$

Also,

$$\mathbb{E}_{a\sim\pi_\theta(\cdot|x)}[\nabla_\theta \log \pi_\theta(a|x)] = 0.$$

Therefore,

$$\begin{aligned}
&\mathbb{E}_{a\sim\pi_\theta(\cdot|x)}\left[A^{\mathrm{GRPO}}(x, a)\nabla_\theta \log \pi_\theta(a|x)\right] \\
&\quad = \frac{\mathbb{E}_{a\sim\pi_\theta(\cdot|x)}[(r(x, a) - p_\theta(x))\nabla_\theta \log \pi_\theta(a|x)]}{\sqrt{p_\theta(x)(1 - p_\theta(x))}} \\
&\quad = \frac{\nabla_\theta p_\theta(x)}{\sqrt{p_\theta(x)(1 - p_\theta(x))}}.
\end{aligned}$$

Finally,

$$\frac{d}{dp}\, 2 \arcsin \sqrt{p} = \frac{1}{\sqrt{p(1 - p)}},$$

so the last display equals

$$\nabla_\theta \left[2 \arcsin \sqrt{p_\theta(x)}\right].$$

□

Remark 10 (Finite groups and degenerate cases)**.** The proof uses population group statistics. For a sampled group, the empirical response distribution replaces $p_\theta(x)$ by $\hat{p} = G^{-1} \sum_i r_i$ and σ by $\sqrt{\hat{p}(1 - \hat{p})}$. As $G \to \infty$, the resulting score update converges to the population expression when $0 < p_\theta(x) < 1$. At finite G its expectation can differ: for $G = 2$, taking zero reward advantage when both rewards agree, the expected group update is $\nabla_\theta p_\theta(x)$, while the population expression is $\nabla_\theta p_\theta(x)/\sqrt{p_\theta(x)(1 - p_\theta(x))}$. The arcsin objective thus motivates the population limit approximated by group normalization.

Remark 11 (Variance-stabilizing intuition)**.** For a Bernoulli reward, A^{GRPO} is the standardized reward and satisfies

$$\mathbb{E}[A^{\mathrm{GRPO}}] = 0, \qquad \mathbb{E}[(A^{\mathrm{GRPO}})^2] = 1.$$

Thus group normalization removes the direct $p_\theta(x)(1 - p_\theta(x))$ reward-variance factor. The transform $2 \arcsin \sqrt{p}$ is the corresponding variance-stabilizing transform for Bernoulli means.

Remark 12 (Score-of-utility view; cf. Corollary 5)**.** The proof above also establishes Corollary 5: the standardized GRPO advantage is exactly $\ell'(p_\theta(x))(r(x, a) - p_\theta(x))$ for $\ell(p) = 2 \arcsin \sqrt{p}$. Reward-side KL shaping changes the population normalization of the combined reward and KL signal, whereas adding the regularizer after outcome normalization preserves the arcsin reward weighting; see Proposition 2.

C MISSING PROOFS

C.1 PROOF OF PROPOSITION 2

Proof. Fix the prompt and suppress x. Write $S_\theta(a) = \nabla_\theta \log \pi_\theta(a)$ and $D_\theta(a) = \log(\pi_\theta(a)/\pi_{\text{ref}}(a))$. Since $\mathbb{E}_{\pi_\theta} S_\theta = 0$, subtracting the population mean of $\tilde{r}$ does not change the expected score update. The reward score identity gives $\mathbb{E}_{\pi_\theta}[rS_\theta] = \nabla_\theta p_\theta$. For the KL term,

$$\nabla_\theta K_\theta = \nabla_\theta \mathbb{E}_{\pi_\theta} D_\theta = \mathbb{E}_{\pi_\theta}[D_\theta S_\theta] + \mathbb{E}_{\pi_\theta}[\nabla_\theta D_\theta] = \mathbb{E}_{\pi_\theta}[D_\theta S_\theta],$$

because $\nabla_\theta D_\theta = S_\theta$ and its expectation is zero. Therefore

$$\mathbb{E}_{\pi_\theta}\left[\frac{\tilde{r} - \mathbb{E}_{\pi_\theta}\tilde{r}}{\tilde{\sigma}} S_\theta\right] = \frac{\nabla_\theta p_\theta - \beta \nabla_\theta K_\theta}{\tilde{\sigma}}.$$

The separate regularization objective $F_\theta = \ell(p_\theta) - \beta K_\theta$ instead has gradient $\ell'(p_\theta)\nabla_\theta p_\theta - \beta \nabla_\theta K_\theta$. In particular, the KL-in-reward update scales both components by the same $\tilde{\sigma}^{-1}$, whereas F_θ applies $\ell'(p_\theta)$ only to the reward component. For binary r, the shaped-reward variance is

$$\tilde{\sigma}^2 = p_\theta(1 - p_\theta) - 2\beta \operatorname{Cov}_{\pi_\theta}(r, D_\theta) + \beta^2 \operatorname{Var}_{\pi_\theta}(D_\theta).$$

Thus normalization also changes with the sampled KL distribution. This proves the stated population direction and its difference from the separately regularized direction. □

Remark 13 (Finite-group amplification of KL noise)**.** The population argument above shows a distorted weighting, not a wrong sign: $\nabla_\theta(p_\theta - \beta K)/\tilde{\sigma}$ is still an ascent direction on the linear objective for each prompt. The empirical collapse is better explained by a finite-group effect. For $\beta > 0$, consider a group of G responses whose verifier rewards are all equal and whose D_i values vary, so that $r_i - \mu_G = 0$ and GRPO assigns zero outcome advantage. Writing $D_i = \sum_j \delta_{i,j}$ and $\bar{D} = \frac{1}{G}\sum_i D_i$, the KL-shaped reward gives $\tilde{r}_i - \tilde{\mu}_G = -\beta(D_i - \bar{D})$ and $\tilde{\sigma}_G = \beta \operatorname{Std}(D)$, hence

$$\hat{A}_i^{\text{KLR}} = -\frac{D_i - \bar{D}}{\operatorname{Std}(D)},$$

which has unit variance across the group and does not depend on β (up to the small constant ε_{std} that implementations add to $\tilde{\sigma}_G$, which is negligible once $\beta \operatorname{Std}(D) \gg \varepsilon_{\text{std}}$). Every token of every response in such a group therefore receives an $O(1)$ update whose direction is set by the sampled within-group KL fluctuation rather than by anything the verifier measured. For binary rewards with group size G, a prompt with pass rate p yields an all-equal group with probability $p^G + (1-p)^G$, which is large for the easy and hard prompts that make up much of a reasoning dataset. The decoupled update in Eq. (11) keeps these groups at zero outcome advantage and adds only the β-scaled future-KL credit, so the injected signal vanishes as $\beta \to 0$. Appendix F.6 measures these quantities in six trained Qwen3-1.7B-Base KL-in-reward runs: about three quarters of the groups are all-equal, and their mean advantage magnitude is about 0.75 at each tested β from 10^{-4} to 1.

C.2 KL ESTIMATOR VALUE UNBIASEDNESS

Proof of Proposition 3. Fix a state s and write $u(a) = \pi_{\text{ref}}(a \mid s)/\pi_\theta(a \mid s)$. Under on-policy sampling $a \sim \pi_\theta(\cdot \mid s)$,

$$\begin{aligned}
\mathbb{E}_{\pi_\theta}\left[\hat{D}^{(k3)}\right] &= \sum_{a \in \mathcal{V}} \pi_\theta(a \mid s)\left[\frac{\pi_{\text{ref}}(a \mid s)}{\pi_\theta(a \mid s)} + \log \frac{\pi_\theta(a \mid s)}{\pi_{\text{ref}}(a \mid s)} - 1\right] \\
&= \sum_{a \in \mathcal{V}} \pi_{\text{ref}}(a \mid s) + \sum_{a \in \mathcal{V}} \pi_\theta(a \mid s) \log \frac{\pi_\theta(a \mid s)}{\pi_{\text{ref}}(a \mid s)} - \sum_{a \in \mathcal{V}} \pi_\theta(a \mid s) \\
&= D_{\text{KL}}[\pi_\theta(\cdot \mid s) \| \pi_{\text{ref}}(\cdot \mid s)].
\end{aligned}$$

□

C.3 On-Policy GRPO with KL Regularization

C.3.1 Derivation of the Regularized Policy-Gradient Weight

Proof. We derive the policy-gradient weight in Eq. (8). Recall the objective function,

$$Q(\theta) = \mathbb{E}_x \ell(\mathbb{E}_{a\sim\pi_\theta} r(x,a)) - \beta \cdot \mathbb{E}_x \mathbb{E}_{a\sim\pi_\theta} \sum_{i=1}^{T} f\left(\frac{\pi_{\text{ref}}(a_i \mid s_i)}{\pi_\theta(a_i \mid s_i)}\right).$$

For a fixed prompt x, denote

$$\begin{aligned} \bar{r}_\theta(x) &:= \mathbb{E}_{a\sim\pi_\theta} r(x,a), \\ u_i &:= \frac{\pi_{\text{ref}}(a_i \mid s_i)}{\pi_\theta(a_i \mid s_i)}, \\ \pi_\theta(a \mid x) &= \prod_{i=1}^{T} \pi_\theta(a_i \mid s_i). \end{aligned}$$

The objective is

$$Q(\theta) = \mathbb{E}_x \ell(\bar{r}_\theta(x)) - \beta \mathbb{E}_x \mathbb{E}_{a\sim\pi_\theta} \sum_{i=1}^{T} f(u_i).$$

Then

$$\nabla_\theta Q(\theta) = \nabla_\theta \mathbb{E}_x \ell(\bar{r}_\theta(x)) - \beta \nabla_\theta \mathbb{E}_x \mathbb{E}_{a\sim\pi_\theta} \sum_{j=1}^{T} f(u_j).$$

First, consider the reward term. By the chain rule,

$$\begin{aligned} \nabla_\theta \ell(\bar{r}_\theta(x)) &= \ell'(\bar{r}_\theta(x)) \nabla_\theta \bar{r}_\theta(x) \\ &= \ell'(\bar{r}_\theta(x)) \nabla_\theta \mathbb{E}_{a\sim\pi_\theta} r(x,a). \end{aligned}$$

Using the likelihood-ratio identity,

$$\begin{aligned} \nabla_\theta \mathbb{E}_{a\sim\pi_\theta} r(x,a) &= \nabla_\theta \sum_a \pi_\theta(a \mid x) r(x,a) \\ &= \sum_a \nabla_\theta \pi_\theta(a \mid x) r(x,a) \\ &= \sum_a \pi_\theta(a \mid x) \nabla_\theta \log \pi_\theta(a \mid x) r(x,a) \\ &= \mathbb{E}_{a\sim\pi_\theta} \left[r(x,a) \nabla_\theta \log \pi_\theta(a \mid x) \right]. \end{aligned}$$

Since

$$\log \pi_\theta(a \mid x) = \sum_{i=1}^{T} \log \pi_\theta(a_i \mid s_i),$$

we have

$$\begin{aligned} \nabla_\theta \mathbb{E}_{a\sim\pi_\theta} r(x,a) &= \mathbb{E}_{a\sim\pi_\theta} \left[r(x,a) \sum_{i=1}^{T} \nabla_\theta \log \pi_\theta(a_i \mid s_i) \right] \\ &= \mathbb{E}_{a\sim\pi_\theta} \sum_{i=1}^{T} r(x,a) \nabla_\theta \log \pi_\theta(a_i \mid s_i). \end{aligned}$$

Therefore,

$$\nabla_\theta \ell(\bar{r}_\theta(x)) = \mathbb{E}_{a\sim\pi_\theta} \sum_{i=1}^{T} \ell'(\bar{r}_\theta(x)) r(x,a) \nabla_\theta \log \pi_\theta(a_i \mid s_i).$$

For any baseline $b(s_i)$,

$$\begin{aligned}
&\mathbb{E}_{a_i\sim\pi_\theta(\cdot|s_i)} \left[b(s_i) \nabla_\theta \log \pi_\theta(a_i \mid s_i) \right] \\
&= b(s_i) \sum_{a_i} \pi_\theta(a_i \mid s_i) \nabla_\theta \log \pi_\theta(a_i \mid s_i) \\
&= b(s_i) \sum_{a_i} \nabla_\theta \pi_\theta(a_i \mid s_i) \\
&= b(s_i) \nabla_\theta \sum_{a_i} \pi_\theta(a_i \mid s_i) \\
&= 0.
\end{aligned}$$

Thus the reward contribution can be written as

$$\nabla_\theta \ell(\bar{r}_\theta(x)) = \mathbb{E}_{a\sim\pi_\theta} \sum_{i=1}^{T} A_i \nabla_\theta \log \pi_\theta(a_i \mid s_i),$$

where

$$\begin{aligned}
A_i &:= \ell'(\bar{r}_\theta(x)) r(x,a) - b(s_i) \\
&= \ell'(\mathbb{E}_{a\sim\pi_\theta} r(x,a))\, r(x,a) - b(s_i).
\end{aligned}$$

Next, consider the regularization term. Define

$$G(\theta; x) := \mathbb{E}_{a\sim\pi_\theta} \sum_{j=1}^{T} f(u_j).$$

Then

$$\begin{aligned}
\nabla_\theta G(\theta; x) &= \nabla_\theta \sum_a \pi_\theta(a \mid x) \sum_{j=1}^{T} f(u_j) \\
&= \sum_a \nabla_\theta \pi_\theta(a \mid x) \sum_{j=1}^{T} f(u_j) + \sum_a \pi_\theta(a \mid x) \nabla_\theta \sum_{j=1}^{T} f(u_j) \\
&= \mathbb{E}_{a\sim\pi_\theta} \left[\left(\sum_{j=1}^{T} f(u_j) \right) \nabla_\theta \log \pi_\theta(a \mid x) + \sum_{j=1}^{T} \nabla_\theta f(u_j) \right].
\end{aligned}$$

For each j,

$$u_j = \frac{\pi_{\text{ref}}(a_j \mid s_j)}{\pi_\theta(a_j \mid s_j)}.$$

Since π_{ref} is fixed,

$$\begin{aligned}
\nabla_\theta u_j &= \nabla_\theta \frac{\pi_{\text{ref}}(a_j \mid s_j)}{\pi_\theta(a_j \mid s_j)} \\
&= -\frac{\pi_{\text{ref}}(a_j \mid s_j)}{\pi_\theta(a_j \mid s_j)^2} \nabla_\theta \pi_\theta(a_j \mid s_j) \\
&= -\frac{\pi_{\text{ref}}(a_j \mid s_j)}{\pi_\theta(a_j \mid s_j)} \nabla_\theta \log \pi_\theta(a_j \mid s_j) \\
&= -u_j \nabla_\theta \log \pi_\theta(a_j \mid s_j).
\end{aligned}$$

Hence

$$\begin{aligned}\nabla_\theta f(u_j) &= f'(u_j)\nabla_\theta u_j \\ &= -f'(u_j)u_j\nabla_\theta \log \pi_\theta(a_j \mid s_j).\end{aligned}$$

Substituting this into $\nabla_\theta G(\theta; x)$ gives

$$\begin{aligned}\nabla_\theta G(\theta; x) &= \mathbb{E}_{a\sim\pi_\theta}\left[\left(\sum_{j=1}^{T} f(u_j)\right)\sum_{i=1}^{T}\nabla_\theta \log \pi_\theta(a_i \mid s_i) - \sum_{j=1}^{T} f'(u_j)u_j\nabla_\theta \log \pi_\theta(a_j \mid s_j)\right] \\ &= \mathbb{E}_{a\sim\pi_\theta}\sum_{i=1}^{T}\left(\sum_{j=1}^{T} f(u_j) - f'(u_i)u_i\right)\nabla_\theta \log \pi_\theta(a_i \mid s_i).\end{aligned}$$

Now we convert the first summation into its causal reward-to-go form. For $j < i$, the quantity $f(u_j)$ is measurable with respect to the history $s_i = (x, a_{<i})$. Therefore,

$$\begin{aligned}&\mathbb{E}_{a\sim\pi_\theta}\left[f(u_j)\nabla_\theta \log \pi_\theta(a_i \mid s_i)\right] \\ &= \mathbb{E}_{s_i}\left[f(u_j)\mathbb{E}_{a_i\sim\pi_\theta(\cdot|s_i)}\left[\nabla_\theta \log \pi_\theta(a_i \mid s_i)\right]\right] \\ &= \mathbb{E}_{s_i}\left[f(u_j)\sum_{a_i}\pi_\theta(a_i \mid s_i)\nabla_\theta \log \pi_\theta(a_i \mid s_i)\right] \\ &= \mathbb{E}_{s_i}\left[f(u_j)\sum_{a_i}\nabla_\theta\pi_\theta(a_i \mid s_i)\right] \\ &= \mathbb{E}_{s_i}\left[f(u_j)\nabla_\theta\sum_{a_i}\pi_\theta(a_i \mid s_i)\right] \\ &= 0.\end{aligned}$$

Thus the past terms $j < i$ can be dropped from the score-function coefficient, and we obtain

$$\nabla_\theta G(\theta; x) = \mathbb{E}_{a\sim\pi_\theta}\sum_{i=1}^{T}\left(\sum_{j\geq i} f(u_j) - f'(u_i)u_i\right)\nabla_\theta \log \pi_\theta(a_i \mid s_i).$$

Finally, since

$$Q(\theta) = \mathbb{E}_x\ell(\bar{r}_\theta(x)) - \beta\mathbb{E}_x G(\theta; x),$$

we have

$$\nabla_\theta Q(\theta) = \mathbb{E}_x\mathbb{E}_{a\sim\pi_\theta}\sum_{i=1}^{T}\left[A_i - \beta\left(\sum_{j\geq i} f(u_j) - f'(u_i)u_i\right)\right]\nabla_\theta \log \pi_\theta(a_i \mid s_i).$$

Therefore, defining

$$w_i := \text{stopgrad}\left(A_i - \beta\left(\sum_{j\geq i} f(u_j) - f'(u_i)u_i\right)\right),$$

we get

$$\nabla_\theta Q(\theta) = \mathbb{E}_x\mathbb{E}_{a\sim\pi_\theta}\sum_{i=1}^{T} w_i\nabla_\theta \log \pi_\theta(a_i \mid s_i).$$

□

C.4 Conditional Policy Improvement with PPO

Consider a finite-horizon MDP with fixed rewards and finite state and action sets. States include time; only reachable states are considered. Let $J(\pi)$ be total expected reward, $\pi = \pi_{\text{old}}$ a full-support rollout policy, $d_\pi(s)$ its visitation probability, and $A_\pi(s,a) = Q^\pi(s,a) - V^\pi(s)$ its true advantage, so $\sum_a \pi(a \mid s) A_\pi(s,a) = 0$. For fixed estimates $\widehat{A}(s,a)$ and $\epsilon \in (0,1)$, define $\rho_{\pi'}(s,a) = \pi'(a \mid s)/\pi(a \mid s)$ and

$$L_\pi^{\text{PPO}}(\pi') = \sum_s d_\pi(s)\, \mathbb{E}_{a\sim\pi(\cdot|s)} \min\Big\{\rho_{\pi'}\widehat{A}, \text{clip}(\rho_{\pi'}, 1-\epsilon, 1+\epsilon)\widehat{A}\Big\}. \tag{12}$$

For integrable random estimates with a fixed conditional law given (s,a), write $\overline{L}_\pi^{\text{PPO}} = \mathbb{E}_{\widehat{A}} L_\pi^{\text{PPO}}$ and $\overline{A}(s,a) = \mathbb{E}[\widehat{A} \mid s,a]$; the expectation is outside the clipped minimum.

Lemma 14 (Single-term gradient saturation from the old policy)**.** *For one fixed estimate $\widehat{A}$, consider isolated ratio-gradient ascent on $\min\{\rho\widehat{A}, \text{clip}(\rho, 1-\epsilon, 1+\epsilon)\widehat{A}\}$:*

$$\rho_0 = 1, \qquad \rho_{k+1} = \max\{0, \rho_k + \eta_k v(\rho_k)\}, \qquad 0 \le \eta_k \le h,$$

where the derivative is chosen as zero at the saturation threshold:

$$v(\rho) = \begin{cases} \widehat{A}\,\mathbf{1}\{\rho < 1+\epsilon\}, & \widehat{A} > 0, \\ \widehat{A}\,\mathbf{1}\{\rho > 1-\epsilon\}, & \widehat{A} < 0, \\ 0, & \widehat{A} = 0. \end{cases}$$

For every k,

$$\begin{cases} 1 \le \rho_k \le 1 + \epsilon + h|\widehat{A}|, & \widehat{A} > 0, \\ \max\{0, 1-\epsilon-h|\widehat{A}|\} \le \rho_k \le 1, & \widehat{A} < 0, \\ \rho_k = 1, & \widehat{A} = 0. \end{cases}$$

The continuous-time flow $\dot\rho = v(\rho)$ almost everywhere from $\rho(0) = 1$ has zero overshoot. These bounds describe one isolated term, not the coupled ratios of a normalized policy.

Proof. The ratio moves toward $1+\epsilon$ or $1-\epsilon$, according to the estimate's sign. Only the crossing step can overshoot, by at most $h|\widehat{A}|$; subsequent derivatives vanish, and projection cannot increase the overshoot. In continuous time, $\rho(t) = 1 + \text{sign}(\widehat{A}) \min\{t|\widehat{A}|, \epsilon\}$. □

Theorem 15 (Conditional policy improvement with PPO)**.** *Under the setup above, the following sufficient conditions hold.*

(a) Bounded wrong-direction movement. *For each state, write $A_a = A_\pi(s,a)$, $p_a = \pi(a \mid s)$, $\rho_a = \pi_+(a \mid s)/p_a$, and*

$$C_s = \{a : A_a \widehat{A}(s,a) > 0\}, \qquad W_s = \{a : A_a \neq 0,\ A_a\widehat{A}(s,a) \le 0\}.$$

Assume the actual updated policy satisfies

$$(\rho_a - 1)\,\text{sign}(A_a) \ge \begin{cases} -\epsilon, & a \in W_s, \\ 0, & a \notin W_s. \end{cases}$$

This is an additional condition, not a consequence of PPO clipping: correct-sign actions cannot move harmfully, and wrong or missed signs have bounded harmful movement. With $G_s = \sum_{a\in C_s} p_a(\rho_a - 1)A_a \ge 0$,

$$J(\pi_+) - J(\pi) \ge \sum_s d_{\pi_+}(s) \left[G_s - \epsilon \sum_{a\in W_s} p_a |A_a| \right].$$

If also $(\rho_a - 1)\,\text{sign}(A_a) \ge \epsilon$ for every $a \in C_s$, then

$$J(\pi_+) - J(\pi) \ge \epsilon \sum_s d_{\pi_+}(s) \left[\sum_{a\in C_s} p_a|A_a| - \sum_{a\in W_s} p_a|A_a| \right].$$

Under this threshold condition, strict improvement follows when

$$\sum_s d_{\pi_+}(s) \sum_{a \in C_s} p_a |A_a| > \tfrac{1}{2} \sum_s d_{\pi_+}(s) \sum_a p_a |A_a|.$$

(b) Local gradient update. *Let $\pi = \pi_{\theta_0}$ belong to a continuously differentiable policy family. Let F have an M-Lipschitz gradient, $M > 0$, on a neighborhood of θ_0 small enough that all ratios lie in $(1-\epsilon, 1+\epsilon)$. For fixed, integrable advantage estimates, assume the population surrogate $\overline{L}(\theta) = \overline{L}_\pi^{\mathrm{PPO}}(\pi_\theta)$ has the correct gradient at the anchor:*

$$g = \nabla \overline{L}(\theta_0) = \nabla F(\theta_0).$$

Here F may be $J(\pi_\theta)$ or a policy-dependent regularized objective. If the segment from θ_0 to $\theta_+ = \theta_0 + \eta g$ stays in the neighborhood, then

$$F(\theta_+) - F(\theta_0) \geq \eta \left(1 - \frac{M\eta}{2}\right) \|g\|^2,$$

which is positive for $g \neq 0$ and $0 < \eta < 2/M$. For a sampled gradient with $\mathbb{E}\widehat{g} = g$ and $\mathbb{E}\|\widehat{g}\|^2 \leq B < \infty$, if its step segments stay in the neighborhood almost surely, then

$$\mathbb{E}[F(\theta_0 + \eta\widehat{g})] - F(\theta_0) \geq \eta \|g\|^2 - \frac{M\eta^2 B}{2},$$

which is positive for $g \neq 0$ and $0 < \eta < 2\|g\|^2/(MB)$.

Proof. **(a)** Since $\sum_a p_a A_a = 0$ and actions outside $C_s \cup W_s$ have zero true advantage,

$$\sum_a \pi_+(a \mid s) A_a = \sum_a p_a(\rho_a - 1) A_a \geq G_s - \epsilon \sum_{a \in W_s} p_a |A_a|.$$

The finite-horizon performance-difference identity gives

$$J(\pi_+) - J(\pi) = \sum_s d_{\pi_+}(s) \sum_a \pi_+(a \mid s) A_\pi(s, a).$$

It follows by telescoping the value terms in $r_t + V^\pi(s_{t+1}) - V^\pi(s_t)$. The threshold condition gives $G_s \geq \epsilon \sum_{a \in C_s} p_a |A_a|$; since C_s, W_s partition the nonzero-advantage actions, the stated majority condition follows.

(b) Full support, continuity, and finiteness ensure an unclipped neighborhood of θ_0. There the clipped and unclipped terms agree for every estimate realization, hence

$$g = \sum_s d_\pi(s) \sum_a \pi(a \mid s) \overline{A}(s, a)\, \nabla_\theta \log \pi_\theta(a \mid s)|_{\theta_0}\,.$$

Conditional unbiasedness for A_π therefore suffices for $F = J$; Eq. (9) supplies the regularized gradient under its population, full-horizon assumptions. Smoothness gives

$$F(\theta_0 + \eta v) - F(\theta_0) \geq \eta \langle g, v \rangle - \tfrac{1}{2} M \eta^2 \|v\|^2.$$

Taking $v = g$, or $v = \widehat{g}$ and then expectations, proves both bounds. □

Part (a) concerns fixed-reward J and assumes control of cross-action effects. Part (b) applies to the regularized objective only when the update gives its correct local gradient; it does not establish unbiasedness of finite-group GRPO estimates. Appendix C.5 analyzes subsequent updates from the same rollout.

C.5 Sequential Mini-Batch Updates from One Rollout

Algorithm 1 takes K sequential mini-batch updates from one rollout, holding its advantages fixed. We extend the local population analysis of Theorem 15(b) to these updates and distinguish staleness from sampling and loss-aggregation effects.

Setup. Write $\theta_0 = \theta_{\text{old}}$, $\pi_0 = \pi_{\theta_0}$, and $P_\theta(\cdot \mid x)$ for the response distribution. We use reverse KL, $f(u) = -\log u$, the full horizon $W = \infty$, and the population group statistics of Proposition 1. With $p_\theta(x) = \mathbb{E}_{P_\theta} r$ and $p_0 = p_{\theta_0}(x)$, define

$$\begin{aligned} Q(\theta) &= \mathbb{E}_x[\ell(p_\theta(x)) - \beta D_{\text{KL}}(P_\theta(\cdot \mid x) \| P_{\text{ref}}(\cdot \mid x))], \\ \hat{A}_i^{(0)} &= \ell'(p_0)(r - p_0) - \beta x_i^{(0)}, \qquad x_i^{(0)} = \sum_{j \geq i} \delta_j^{(0)}, \\ \delta_j^{(0)} &= \log \frac{\pi_0(a_j \mid s_j)}{\pi_{\text{ref}}(a_j \mid s_j)}, \qquad \rho_i(\theta) = \frac{\pi_\theta(a_i \mid s_i)}{\pi_0(a_i \mid s_i)}. \end{aligned}$$

For every prefix-dependent baseline $b(s_i)$, normalization of $\pi_\theta(\cdot \mid s_i)$ gives

$$\begin{aligned} \mathbb{E}_{a_i \sim \pi_0(\cdot \mid s_i)}[(\rho_i(\theta) - 1) b(s_i)] = 0, \\ \mathbb{E}_{a_i \sim \pi_0(\cdot \mid s_i)}[\nabla_\theta \rho_i(\theta) b(s_i)] = 0. \end{aligned} \tag{13}$$

The fixed-anchor population surrogates are

$$\begin{aligned} L(\theta) &= Q(\theta_0) + \mathbb{E}_x \mathbb{E}_{P_0} \sum_i (\rho_i(\theta) - 1) \hat{A}_i^{(0)}, \\ L^{\text{clip}}(\theta) &= \mathbb{E}_x \mathbb{E}_{P_0} \sum_i \psi(\rho_i(\theta), \hat{A}_i^{(0)}), \end{aligned} \tag{14}$$

where $\psi(\rho, A) = \min\{\rho A, \text{clip}(\rho, 1-\epsilon, 1+\epsilon)A\}$ for $A \geq 0$, and for $A < 0$ the dual clip replaces this minimum by its maximum with $\bar{c}A$, with $\epsilon \in (0, 1)$ and $\bar{c} > 1+\epsilon$ ($\epsilon = 0.2$, $\bar{c} = 3$ in our runs; $\bar{c} = \infty$ gives standard PPO). We omit the action-independent $+1$ in the boundary term $\delta_i + 1$: subtracting β from every advantage leaves L and its gradient unchanged by Eq. (13). The implementation retains this constant, whose effect on the population clipped gradient also cancels when all ratios are inside the clip interval; outside that region, the shift can change advantage signs and gates.

Lemma 16 (Clipping gates tokens without rescaling)**.** *For a sampled token with frozen advantage A, away from the thresholds $1 \pm \epsilon$ and $\bar{c}$,*

$$\nabla_\theta \psi(\rho_\theta, A) = A \rho_\theta \mathbf{1}[\textit{token} \notin \mathcal{G}] \nabla_\theta \log \pi_\theta(a \mid s),$$

where

$$\begin{aligned} \mathcal{G} = \{A > 0, \rho_\theta > 1 + \epsilon\} \cup \{A < 0, \rho_\theta < 1 - \epsilon\} \\ \cup \{A < 0, \rho_\theta > \bar{c}\}. \end{aligned}$$

Thus each ungated token keeps the relative weights of its verifier and future-KL terms. At fixed ratios, the future-KL term affects gating only through the sign of A; at θ_0, all ratios equal one and $\nabla L^{\text{clip}}(\theta_0) = \nabla L(\theta_0)$.

Proof. As in Lemma 14, $\psi = A\rho$ except on the gated intervals, where it is constant: $A(1+\epsilon)$ for positive A, and $A(1-\epsilon)$ or $\bar{c}A$ for negative A. Differentiating each interval gives the result. □

Proposition 17 (Population updates on a fixed-anchor surrogate)**.** *Assume (A1) finitely many prompts, a finite vocabulary, response length at most T, and a full-support reference policy; (A2) $\log \pi_\theta(a \mid s)$ is twice continuously differentiable on a compact convex neighborhood B of θ_0, with $\|\nabla_\theta \log \pi_\theta(a \mid s)\| \leq L_1$; and (A3) $p_\theta(x) \in [p_{\min}, 1 - p_{\min}]$ on B for some $p_{\min} > 0$, with ℓ twice continuously differentiable on this interval. Then:*

(i) $L(\theta_0) = Q(\theta_0)$ and $\nabla L(\theta_0) = \nabla Q(\theta_0)$.

(ii) For $\theta \in B$, $\nabla L(\theta) = \mathbb{E}_x \mathbb{E}_{P_0} \sum_i \rho_i(\theta) \hat{A}_i^{(0)} \nabla_\theta \log \pi_\theta(a_i \mid s_i)$. If $L_1 \|\theta - \theta_0\| < \log(1+\epsilon)$, no token is gated and $\nabla L^{\text{clip}}(\theta) = \nabla L(\theta)$, so the population clipped updates are gradient steps on the same L.

(iii) For $\theta \in B$,

$$\begin{aligned} Q(\theta) - L(\theta) = \mathbb{E}_x \Big[\tfrac{1}{2} \ell''(\xi_x)(p_\theta(x) - p_0(x))^2 + \ell'(p_0(x)) R_r(\theta; x) \\ - \beta R_{\delta^{(0)}}(\theta; x) - \beta D_{\text{KL}}(P_\theta(\cdot \mid x) \| P_0(\cdot \mid x)) \Big], \end{aligned} \tag{15}$$

where ξ_x lies between $p_0(x)$ and $p_\theta(x)$, and for a fixed per-token cost c,

$$R_c(\theta; x) = \sum_i \big(\mathbb{E}_{s_i \sim d_i^\theta} - \mathbb{E}_{s_i \sim d_i^0}\big)\mathbb{E}_{a \sim \pi_\theta(\cdot|s_i)} A_c^{\pi_0}(s_i, a).$$

Here d_i^θ is the prefix visitation distribution under P_θ, $A_c^{\pi_0}$ is the cost advantage under π_0, and the two costs are the terminal reward r and the frozen log-ratios $\delta^{(0)}$. All four terms and their gradients vanish at θ_0.

(iv) With $M_\Delta = \sup_{\theta \in B} \|\nabla^2(Q - L)(\theta)\| < \infty$,

$$\|\nabla L(\theta_k) - \nabla Q(\theta_k)\| \le M_\Delta \|\theta_k - \theta_0\|, \qquad \theta_k \in B.$$

(v) If ∇Q is M_Q-Lipschitz on B and the population steps $\theta_{k+1} = \theta_k + \eta \nabla L(\theta_k)$ remain in B, with $\|\nabla L(\theta_k)\| \le G$ and $\eta > 0$, then

$$Q(\theta_K) - Q(\theta_0) \ge \eta \sum_{k=0}^{K-1} \|\nabla L(\theta_k)\|^2 - \eta^2 G^2 \left[\frac{M_Q K}{2} + \frac{M_\Delta K(K-1)}{2}\right].$$

The bound also holds for population clipped steps under (ii). For $K = 1$ and $G = \|\nabla L(\theta_0)\|$, it is Theorem 15(b) with $F = Q$ and $M = M_Q$.

Proof. **(iii) and (i).** Fix a prompt and a parameter-independent cost c, with $J_c(\theta) = \mathbb{E}_{P_\theta} \sum_i c_i$ and $C_i = \sum_{j \ge i} c_j$. The performance-difference identity gives

$$J_c(\theta) - J_c(\theta_0) = \sum_i \mathbb{E}_{d_i^\theta} \mathbb{E}_{\pi_\theta} A_c^{\pi_0} = \mathbb{E}_{P_0} \sum_i (\rho_i(\theta) - 1) C_i + R_c(\theta; x).$$

The second equality uses $\mathbb{E}_{\pi_0} A_c^{\pi_0}(s_i, \cdot) = 0$, $\mathbb{E}_{P_0}[C_i \mid s_i, a_i] = A_c^{\pi_0}(s_i, a_i) + V_c^{\pi_0}(s_i)$, and Eq. (13). Apply this identity first to $c = r$ (terminal, so $J_r = p_\theta$ and $C_i = r$), and then to $c = \delta^{(0)}$ (so $C_i = x_i^{(0)}$). Taylor expansion of $\ell(p_\theta)$ supplies the quadratic term, while

$$D_{\mathrm{KL}}(P_\theta \| P_{\mathrm{ref}}) = J_{\delta^{(0)}}(\theta) + D_{\mathrm{KL}}(P_\theta \| P_0)$$

follows from $\log(P_\theta / P_{\mathrm{ref}}) = \log(P_0 / P_{\mathrm{ref}}) + \log(P_\theta / P_0)$. Combining these identities and centering r by p_0 yields Eq. (15). At θ_0, the Taylor remainder and the KL term have zero value and gradient. Each R_c is a sum of products $(d_i^\theta(s) - d_i^0(s))\mathbb{E}_{\pi_\theta} A_c^{\pi_0}(s, \cdot)$ whose two factors vanish there, proving (iii) and hence (i).

(ii) and (iv). Differentiate the frozen-advantage surrogate. The score bound gives $|\log \rho_i(\theta)| \le L_1 \|\theta - \theta_0\|$; under (ii), all ratios lie in $(1-\epsilon, 1+\epsilon)$ because $\log(1+\epsilon) \le -\log(1-\epsilon)$. The clipping lemma proves gradient equality. Under (A1)–(A3), $Q - L$ is twice continuously differentiable on compact B; integrating its Hessian along the segment from θ_0 to θ_k and using (i) proves (iv).

(v). Let $g_k = \nabla L(\theta_k)$. Smoothness and (iv) give

$$Q(\theta_{k+1}) - Q(\theta_k) \ge \eta \|g_k\|^2 - \eta M_\Delta \|\theta_k - \theta_0\| \|g_k\| - \tfrac{1}{2} M_Q \eta^2 \|g_k\|^2.$$

Use $\|\theta_k - \theta_0\| \le \eta k G$ and sum over $k < K$. □

Interpretation and finite mini-batches. Eq. (15) separates the frozen reward gain, prefix visitation mismatch, and the omitted proximal term $-\beta D_{\mathrm{KL}}(P_\theta \| P_0)$. Recomputing log-ratios approximately restores the latter's gradient near θ_0; Appendix F.12 measures this effect. The population staleness error vanishes at θ_0 and contributes the $O(\eta^2 K(K-1))$ term in (v). In contrast, a local loss-side penalty generally omits the score-function terms $\sum_{j \ge i} f(u_j)$ even at the anchor (Proposition 6); reducing the step size alone does not remove that population gradient mismatch.

Within the unclipped neighborhood of (ii), stochastic updates $\theta_{k+1} = \theta_k + \eta \hat{g}_k$ require the additional condition $\mathbb{E}[\hat{g}_k \mid \mathcal{F}_k] = c \nabla L(\theta_k)$ for unbiasedness, where $\mathcal{F}_k$ contains the previous updates and $c > 0$ is a fixed normalization. Equal-sized batches of independent rollout samples, grouped independently of their contents, with population advantage statistics and a fixed denominator, suffice;

exchangeability alone does not. Sampling noise, finite-group statistics, and response-dependent partitioning or normalization are separate from the population bound. In particular, the first mini-batch has no staleness, but need not equal the population gradient.

If mini-batch k has N_k valid tokens and $N = \sum_k N_k$, equal averaging of its token-mean loss gives each token weight $N/(KN_k)$ relative to the full-rollout token mean. This reweighting disappears for $K = 1$ or equal N_k, not merely for a random partition independent of the responses. At a fixed parameter, weighting mini-batch means by N_k/N exactly recovers the rollout token mean; sequential updates at different parameters do not obey that identity. Random token denominators also need not yield unbiased population gradients.

C.6 KL in Reward from the Perspective of Optimality

This subsection states and proves the optimality view summarized in Section 5. The same future-KL structure also appears from the optimality conditions of entropy-regularized RL. For a fixed prompt, consider

$$Q(\pi) = \mathbb{E}_{a\sim\pi(\cdot|x)}[r(x,a)] - \beta\, D_{\mathrm{KL}}[\pi(\cdot|x)\|\pi_{\mathrm{ref}}(\cdot|x)]. \tag{16}$$

Proposition 18 (Optimal policy)**.** *The maximizer satisfies* $\pi^*(a \mid x) \propto \pi_{\mathrm{ref}}(a \mid x)\exp(r(x,a)/\beta)$.

Proposition 19 (Constant shifted reward)**.** *Equivalently, for each prompt* x *there exists a constant* $C(x)$ *such that*

$$r(x,a) - \beta \log \frac{\pi^*(a \mid x)}{\pi_{\mathrm{ref}}(a \mid x)} = C(x). \tag{17}$$

Because autoregressive likelihood ratios decompose over tokens, Eq. (17) implies the following process-reward form, where $a^{(j)} = (a_1, a_2, \cdots, a_j)$ denotes the tokens generated so far and $a^{(-j)} = (a_{j+1}, a_{j+2}, \cdots, a_T)$ the remaining tokens.

Proposition 20 (Future KL as process reward)**.** *At the optimal policy, the entropy-regularized process reward after prefix* $a^{(t)}$ *can be written as*

$$r_t(x, a^{(t)}) = r(x,a) - \beta \sum_{j=t+1}^{T} \log \frac{\pi^*(a_j \mid x, a^{(j-1)})}{\pi_{\mathrm{ref}}(a_j \mid x, a^{(j-1)})}. \tag{18}$$

The proofs are given below. This optimality view is not needed to derive Theorem 4, but it gives the same interpretation: KL regularization is a process-level signal, and the relevant token weight is a future cost-to-go rather than an isolated per-token penalty.

The optimality view also makes the causal structure explicit: adjacent prefixes satisfy $r_t - r_{t-1} = \beta \log(\pi^*(a_t \mid x, a^{(t-1)})/\pi_{\mathrm{ref}}(a_t \mid x, a^{(t-1)}))$, so the token log-ratio is exactly the change in process reward. The future-KL correction in FRPO is a practical on-policy approximation to this structure, using sampled log-ratios and GRPO's group-normalized advantage in place of a learned process value.

C.6.1 Proof of Proposition 18

Proof. Consider for a fixed prompt $x \in \mathcal{V}^*$, and our optimization objective is:

$$\begin{aligned}
&\max_{\pi(\cdot|x)} \{\mathbb{E}_{a\sim\pi(\cdot|x)}[r(x,a)] - \beta D_{\mathrm{KL}}(\pi\|\pi_{\mathrm{ref}})\} \\
&= \max_{\pi(\cdot|x)} \left\{ \sum_{a\in\mathcal{V}^*} \pi(a|x) \cdot \left(r(x,a) - \beta \ln \left(\frac{\pi(a|x)}{\pi_{\mathrm{ref}}(a|x)} \right) \right) \right\},
\end{aligned}$$

under the condition $\sum_{a\in\mathcal{V}^*} \pi(a|x) = 1$.

Thus, we construct the Lagrangian of the above optimization problem,

$$\mathcal{L}(\pi(\cdot|x), \lambda) = \sum_{a\in\mathcal{V}^*} \pi(a|x) \cdot \left(r(x,a) - \beta \ln \left(\frac{\pi(a|x)}{\pi_{\mathrm{ref}}(a|x)} \right) \right) - \lambda \Big(1 - \sum_{a\in\mathcal{V}^*} \pi(a|x) \Big).$$

Setting the partial derivative of $\mathcal{L}(\pi, \lambda)$ with respect to π equal to 0,

$$\frac{\partial}{\partial \pi(a|x)} \mathcal{L}(\pi, \lambda) = 0,$$

we have

$$\begin{aligned}
&\frac{\partial}{\partial \pi(a|x)} \left[\sum_{a \in \mathcal{V}^*} \pi(a|x) \cdot \left(r(x,a) - \beta \ln \left(\frac{\pi(a|x)}{\pi_{\text{ref}}(a|x)} \right) \right) - \lambda \left(1 - \sum_{a \in \mathcal{V}^*} \pi(a|x) \right) \right] = 0 \\
\Leftrightarrow\ & r(x,a) - \beta \left(\ln \left(\frac{\pi(a|x)}{\pi_{\text{ref}}(a|x)} \right) + 1 \right) - \lambda = 0 \\
\Leftrightarrow\ & \frac{1}{\beta} \cdot r(x,a) - \frac{1}{\beta} \cdot \lambda - 1 = \ln \left(\pi(a|x) \right) - \ln \left(\pi_{\text{ref}}(a|x) \right) \\
\Leftrightarrow\ & \ln \pi(a|x) = \ln \pi_{\text{ref}}(a|x) + \frac{1}{\beta} \cdot r(x,a) - \frac{1}{\beta} \cdot \lambda - 1 \\
\Leftrightarrow\ & \pi(a|x) = \pi_{\text{ref}}(a|x) \exp \left(\frac{1}{\beta} \cdot r(x,a) - \frac{1}{\beta} \lambda - 1 \right) \propto \pi_{\text{ref}}(a|x) \exp \left(\frac{1}{\beta} r(x,a) \right).
\end{aligned}$$

Therefore, we complete the proof. □

C.6.2 Proof of Proposition 20

Proof. From Proposition 19, under optimal policy π^* and reward r:

$$C = r(x,a) - \beta \ln \frac{\pi^*(a|x)}{\pi_{\text{ref}}(a|x)}.$$

Given the auto-regressive generation property of language modeling, we have:

$$\ln \frac{\pi^*(a|x)}{\pi_{\text{ref}}(a|x)} = \ln \frac{\prod_{j=1}^{T} \pi^*(a_j|x, a^{(j-1)})}{\prod_{j=1}^{T} \pi_{\text{ref}}(a_j|x, a^{(j-1)})} = \sum_{j=1}^{T} \ln \frac{\pi^*(a_j|x, a^{(j-1)})}{\pi_{\text{ref}}(a_j|x, a^{(j-1)})}.$$

Therefore, we have:

$$C = r(x,a) - \beta \sum_{j=1}^{t} \ln \left(\frac{\pi^*(a_j|x, a^{(j-1)})}{\pi_{\text{ref}}(a_j|x, a^{(j-1)})} \right) - \beta \sum_{j=t+1}^{T} \ln \left(\frac{\pi^*(a_j|x, a^{(j-1)})}{\pi_{\text{ref}}(a_j|x, a^{(j-1)})} \right)$$

Therefore,

$$\begin{aligned}
& r_t(x, a^{(t)}) \\
=\ & \mathbb{E}_{a^{(-t)} \sim \pi^*(\cdot|x, a^{(t)})} \left[r(x,a) - \beta \sum_{j=t+1}^{T} \ln \left(\frac{\pi^*(a_j|x, a^{(j-1)})}{\pi_{\text{ref}}(a_j|x, a^{(j-1)})} \right) \right] \\
=\ & \mathbb{E}_{a^{(-t)} \sim \pi^*(\cdot|x, a^{(t)})} \left[C + \beta \sum_{j=1}^{t} \ln \left(\frac{\pi^*(a_j|x, a^{(j-1)})}{\pi_{\text{ref}}(a_j|x, a^{(j-1)})} \right) \right] \\
=\ & C',
\end{aligned}$$

for some constant $C' \in \mathbb{R}$, where the final equation holds due to the fact that $a^{(t)}$ is fixed given the conditional distribution $\pi^*(\cdot|x, a^{(t)})$. Thus, the process reward $r_t(x, a^{(t)})$ should be a constant no matter what the future path is under the optimal policy π^*. Therefore, given the whole reasoning trajectory $a = (a^1, \cdots, a^T)$, the universal equality holds:

$$r_t(x, a^{(t)}) = r(x,a) - \beta \sum_{j=t+1}^{T} \ln \frac{\pi^*(a_j|x, a^{(j-1)})}{\pi_{\text{ref}}(a_j|x, a^{(j-1)})}.$$

□

Algorithm 1 Future-KL Regularized Policy Optimization (FRPO)

Require: Policy π_θ, reference policy π_{ref}, rollout batch $\mathcal{B}$, mini-batch size B_{mini}, group size G, KL coefficient β, horizon $W \in \mathbb{N} \cup \{\infty\}$, PPO clip ϵ

for each training iteration **do**
  Set rollout policy $\pi_{\theta_{\text{old}}} \leftarrow \pi_\theta$
  for each prompt $x \in \mathcal{B}$ **do**
    Sample responses $\{a_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot|x)$
    Evaluate rewards $r_i = r(x, a_i)$ and token masks $m_{i,t}$
    Compute $\mu_G = \frac{1}{G}\sum_i r_i$ and $\sigma_G = \sqrt{\frac{1}{G-1}\sum_i (r_i - \mu_G)^2}$ ▷ Unbiased, as in verl
    $\hat{A}_i^{\text{GRPO}} \leftarrow (r_i - \mu_G)/(\sigma_G + \varepsilon_{\text{std}})$
    for each response token t of response i **do**
      $\delta_{i,t} \leftarrow \log \pi_{\theta_{\text{old}}}(a_{i,t}|x, a_{i,<t}) - \log \pi_{\text{ref}}(a_{i,t}|x, a_{i,<t})$ ▷ Trainer log-probs at θ_{old}
      $S_{i,t}^{(W)} \leftarrow \sum_{j=t}^{\min\{t+W-1, T_i\}} m_{i,j}\delta_{i,j}$ ▷ Windowed KL-to-go
      $\hat{A}_{i,t}^{FRPO(W)} \leftarrow \text{sg}(\hat{A}_i^{\text{GRPO}} - \beta S_{i,t}^{(W)})$
    end for
  end for
  Split the rollout batch into actor mini-batches of size B_{mini}
  Update θ once per mini-batch by averaging the following PPO term over valid response tokens:

$$L_{i,t}^{\text{PPO}}(\theta) = \min\Big(\rho_{i,t}\hat{A}_{i,t}^{FRPO(W)}, \text{clip}(\rho_{i,t}, 1-\epsilon, 1+\epsilon)\hat{A}_{i,t}^{FRPO(W)}\Big),$$

  where $\rho_{i,t} = \pi_\theta(a_{i,t}|x, a_{i,<t})/\pi_{\theta_{\text{old}}}(a_{i,t}|x, a_{i,<t})$.
end for

D Algorithmic Details for FRPO

Implementation notes. The default future term uses the sampled log-ratio ($k1$) estimator. A $k3$ variant replaces $\delta_{i,j}$ for the future tokens $j > t$ in $S_{i,t}$ by $-\log u_{i,j} + u_{i,j} - 1$, while the current token keeps the boundary term $f(u_t) - f'(u_t)u_t = -\log u_{i,t} = \delta_{i,t}$, where $u_{i,j} = \pi_{\text{ref}}(a_{i,j}|s_{i,j})/\pi_{\theta_{\text{old}}}(a_{i,j}|s_{i,j})$. All sums are masked over response tokens and exclude prompt and padding tokens; the default full-horizon setting uses $W = \infty$, and every finite window includes the current response token. The log-probabilities in $\delta_{i,t}$ are recomputed by the trainer at θ_{old} after rollout and before any actor update, rather than taken from the inference engine. Unless stated otherwise, the future-KL term is computed once per rollout batch and held fixed across its mini-batches; the alternative recomputes it under the current policy before each actor update, which Appendices F.7 and F.8 use. For $k1$, the implementation also adds the action-independent constant $+1$ of the boundary term $f(u_t) - f'(u_t)u_t = \delta_t + 1$ to $S_{i,t}$, which shifts every advantage by $-\beta$. In our experiments, PPO epochs are set to 1: a train batch is rolled out once for efficiency and then split into actor mini-batches, each used for a single update. The first mini-batch therefore has $\rho_{i,t} \equiv 1$ (no staleness), and each later one is updated at the parameters reached after the preceding mini-batches. Appendix C.5 analyzes these sequential updates, and Appendix F.12 measures their staleness and compares the fixed and recomputed future-KL terms. For negative advantages, the implementation also applies dual clipping with $\bar{c} = 3$, which zeroes the gradient of tokens with $\rho_{i,t} > \bar{c}$. Before the split, verl's batch balancing orders the sequences on each data-parallel rank by length for load balancing (actor-side shuffling is disabled), and the token-mean loss is normalized within each mini-batch. Refer to Algorithm 1 for details.

E Discussion about Future KL

E.1 Why Not Put KL into the Reward Before Normalization?

Section 5 uses a decoupled update: compute the GRPO outcome advantage first, then add a token-wise future-KL correction. There are three natural ways to combine future KL with GRPO-style normalization.

Table 4: Where this paper sits among KL-regularized policy-gradient analyses. Linear ℓ allows reward and regularizer to fuse into a single augmented reward; under GRPO's non-linear $\ell(p) = 2\arcsin\sqrt{p}$ they cannot, and the right insertion point for the future-KL correction is on the regularizer side of the ℓ'-asymmetry.

Setting	Prompt objective	Reward+KL fuseable?	Future-reg in PG?	Insertion point
MaxEnt RL / soft-Q	$\mathbb{E}_\pi[r] - \beta\, D_{\mathrm{KL}}$	yes (augmented reward)	yes (textbook)	either
PPO + KL-in-reward (+GAE)	$\mathbb{E}_\pi[r] - \beta\, D_{\mathrm{KL}}$	yes	yes (via value)	reward shaping
GRPO (this paper)	$\boldsymbol{\ell(\mathbb{E}_\pi[r]) - \beta\, D_{\mathrm{KL}}}$	**no** (Prop. 2)	**yes (Thm. 4)**	**after advantage**

(a) **Decoupled:** Compute GRPO advantage from the original outcome reward, then add the future-KL correction:

$$\hat{A}_{i,t} = \frac{r_i - \mu_G}{\sigma_G} - \beta \cdot \sum_{j=t}^{T} \delta_{i,j}.$$

(b) **Coupled, trajectory-level:** Augment the outcome reward with the *full* trajectory KL, then apply group normalization:

$$\tilde{r}_i = r_i - \beta \cdot \sum_{j=1}^{T} \delta_{i,j}, \qquad \tilde{A}_i = \frac{\tilde{r}_i - \tilde{\mu}_G}{\tilde{\sigma}_G}.$$

This yields a *uniform* advantage for all tokens in response i—the per-token differentiation from future KL is lost entirely. It also changes the group mean and standard deviation, so the regularizer affects not only policy drift but also the scale and ordering used to construct the GRPO advantage.

(c) **Coupled, step-level:** Use the step-dependent process reward $r_{i,t}$ and normalize across the group at each step:

$$\hat{A}_{i,t} = \frac{r_{i,t} - \mu_{G,t}}{\sigma_{G,t}}, \qquad \mu_{G,t} = \frac{1}{G}\sum_{k=1}^{G} r_{k,t}, \quad \sigma_{G,t} = \mathrm{std}_k(r_{k,t}).$$

In this strategy, $r_{i,t}$ uses token-level KL instead of trajectory-level KL, leading to possibly higher variance. In addition, trajectories have different lengths, so at large t only a subset of trajectories contributes, further degrading the estimates.

Analysis. Strategy (b) defeats the purpose of process rewards: by collapsing the future KL into a single trajectory-level scalar, all tokens share the same advantage, and the fine-grained credit assignment from Proposition 20 is discarded. Moreover, Proposition 2 shows that population KL-in-reward normalization scales both the reward and KL gradients by the shaped-reward standard deviation, whereas the separately regularized arcsin objective scales only the reward gradient by $\ell'(p)$. In addition, Remark 13 shows that in every group with identical verifier rewards and varying KL values the KL-in-reward advantage reduces to a unit-variance function of within-group KL noise in which β cancels; in six trained Qwen3-1.7B-Base KL-in-reward runs such groups make up about three quarters of all groups at each tested β from 10^{-4} to 1 (Appendix F.6). Strategy (c) is closer to the process-reward interpretation, but suffers from statistical instability: with small groups, the per-step normalization constants $\mu_{G,t}$ and $\sigma_{G,t}$ are noisy estimates, and variable trajectory lengths mean that for large t, very few trajectories contribute to the statistics. This can lead to erratic advantage values in the later portion of responses.

Strategy (a), used by FRPO, avoids both problems by **separating the two roles** of the advantage:

- *Trajectory ranking* is handled by $\hat{A}_i^{\mathrm{GRPO}}$, which benefits from stable group normalization on outcome rewards. Crucially, since the original reward enters the normalization unchanged, the implicit $2\arcsin(\sqrt{p})$ objective is preserved.
- *Token differentiation* is handled by the future KL term $-\beta \cdot \sum_{j=t}^{T} \delta_{i,j}$, which operates within each trajectory and does not require cross-trajectory normalization.

F More Experiment Details

We show the experiment configurations in Table 5 and Table 6. We use vLLM (Kwon et al., 2023) as the inference backend to speedup rollout. Most experiments are conducted on 4×8 NVIDIA H20 GPU nodes.

Table 5: Hyperparameters for Baselines. The KL coefficients are shared across models; Appendix F.1 explains how they were set.

Hyperparameters	**Qwen3-1.7B-Base**	**Qwen3-30B-A3B-Base**
Learning Rate	1e-6	1e-6
PPO Epochs	1	1
Max Prompt Length	2048	2048
Max Response Length	8192	8192
Train Batch Size	256	256
PPO Mini Batch Size	64	64
Rollout Temperature	1.0	1.0
Group Size	5	5
Actor Loss Aggregation	token mean	token mean
$k3$ KL-loss coefficient	10^{-3}	10^{-3}
KL-in-reward coefficient	10^{-5}	10^{-5}

Table 6: Hyperparameters for FRPO. β was selected once in a preliminary sweep and then fixed for all models (Appendix F.1).

Hyperparameters	**Qwen3-1.7B-Base**	**Qwen3-30B-A3B-Base**
Learning Rate	1e-6	1e-6
PPO Epochs	1	1
Max Prompt Length	2048	2048
Max Response Length	8192	8192
Train Batch Size	256	256
PPO Mini Batch Size	64	64
Rollout Temperature	1.0	1.0
Group Size	5	5
Actor Loss Aggregation	token mean	token mean
β	0.01	0.01

For math reasoning, we add an additional prompt behind each problem, as shown in F.1.

F.1: Prompt for math reasoning

{problem}

Let's think step by step and output the final answer within \boxed{}.

F.1 Choice of Regularization Coefficients

The three KL placements in Table 1 and Figure 2 are each run at a fixed coefficient that is shared across models. This subsection explains how each coefficient was set and compares the placements at matched coefficients.

How each coefficient was set. FRPO uses $\beta = 10^{-2}$ in every experiment unless a sweep is stated. We selected it in a preliminary sweep on Qwen3-4B-Base over $\beta \in \{0.1, 0.05, 10^{-2}, 2 \times 10^{-3}\}$, where $\beta = 10^{-2}$ gave the highest accuracy (83.9%, against 75.7%, 80.9% and 80.5%), and kept it fixed for all models. The loss-side coefficient 10^{-3} is the default `kl_loss_coef` of verl (Sheng

et al., 2024) and the value used in most of its GRPO example scripts, so the loss-side baseline follows the standard framework setting. KL-in-reward uses the small coefficient 10^{-5}. Its failure does not depend on this value: in all-equal groups the coefficient cancels under group normalization (Remark 13), and GRPO with KL in the reward stays far below the other methods at every $\beta \in \{10^{-4}, 10^{-3}, 10^{-2}, 0.1, 0.5, 1\}$ (Appendix F.6).

Matched coefficients at 1.7B. For FRPO and the loss-side penalty, the same β multiplies the same KL term of the regularized objective, so the sweep of Appendix F.8 compares the two placements at matched coefficients, $\beta \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$. It shows that the default $\beta = 10^{-2}$ is the best of the four values for FRPO. And $\beta = 10^{-3}$ works better on pass@16 for loss-side KL.

F.2 Multi-Seed Results at Matched Training Steps

We compare FRPO (reverse KL, $k1$), GRPO with the loss-side $k3$ penalty, and GRPO without KL regularization on Qwen3-1.7B-Base across independent runs. Every run is taken at the same training length of 200 steps and evaluated offline with the same protocol: 16 samples per problem, temperature 1.0, and an 8,192-token budget. In those runs FRPO recomputes the future KL under the current policy, which Appendix F.12 finds makes no detectable difference. Table 7 reports the mean and sample standard deviation across runs; response length and entropy are rollout statistics averaged over the last twenty training steps. Each method uses the coefficient of Appendix F.1.

Table 7: MATH500 accuracy (%) on Qwen3-1.7B-Base for the multi-seed training, with performance reported by mean ± sample SD across independent runs.

Method	pass@16	mean@16	Resp. len.	Entropy
FRPO	$\mathbf{89.43 \pm 0.48}$	66.72 ± 0.75	2839	0.122
GRPO + loss-side $k3$ penalty	87.11 ± 0.38	$\mathbf{67.78 \pm 0.15}$	1419	0.049
GRPO without KL	85.93 ± 1.03	65.99 ± 0.34	1538	0.039

At matched training steps, FRPO attains the highest pass@16, 2.3 and 3.5 points above the loss-side penalty and GRPO without KL, with mean@16 within 1.1 points of the loss-side penalty and 0.7 points above GRPO without KL. It also retains 2.5 times the entropy of the loss-side run, consistent with the role of the regularizer in preserving exploration.

F.3 KL Direction and Estimator Ablations

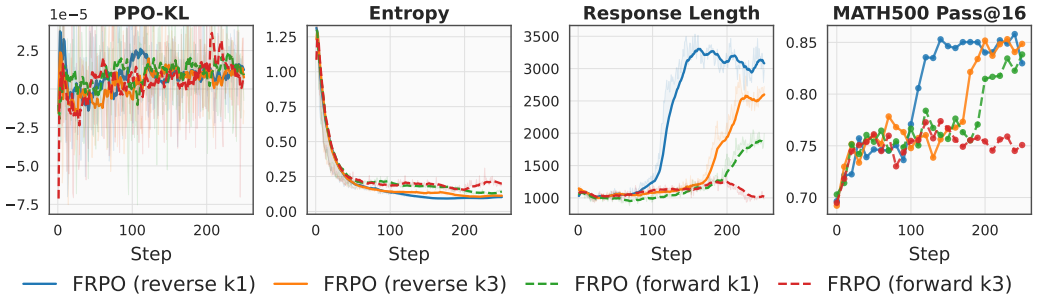


Figure 3: Reverse KL versus forward KL, with $k1$ and $k3$ as the estimators on Qwen3-1.7B-Base.

Forward KL versus Reverse KL. Forward KL $D_{\mathrm{KL}}(\pi_{\mathrm{ref}} \| \pi_\theta)$ is mode-covering: it penalizes the current policy for assigning too little probability to regions supported by the reference policy. Reverse KL $D_{\mathrm{KL}}(\pi_\theta \| \pi_{\mathrm{ref}})$ is more mode-seeking: it discourages generations that are unlikely under the reference. Figure 3 shows that reverse KL works better in our setting, likely because forward KL keeps entropy higher than is useful for math reasoning.

Different KL Estimators. Many RL frameworks favor $k3$ over $k1$ because both are value-unbiased for reverse KL under on-policy sampling, while $k3$ is always nonnegative. In FRPO, however, the

future-KL term is used as a signed credit-assignment signal rather than only as a scalar divergence monitor. The nonnegativity of $k3$ makes its cumulative future term one-sided: it always pushes the token weight in the penalty direction, even when a centered signal would be needed to distinguish relatively helpful and harmful deviations within a group. Figure 3 shows a gap between $k1$ and $k3$ estimators, especially on forward KL, so we use $k1$ as the default estimator in FRPO.

F.4 Subtracting a Baseline for $k3$ Estimators

Due to the nonnegativity of $k3$ estimators, the future KL term will always be a penalty, therefore pushing the policy towards the same direction. To avoid such a phenomenon, we implement a naive baseline, which takes the average KL divergence at the token level among all tokens in the same batch, and then scaled by the remaining length of a particular response from the current token i, i.e.,

$$b_i = (T - i) \cdot \overline{D_{\mathrm{KL}}}.$$

In Table 8, we summarize the results of incorporating baseline subtraction into the $k3$ estimators for both forward and reverse KL. We observe that baseline subtraction alone does not lead to performance improvements. We hypothesize that this is because the estimated baselines are relatively coarse and do not accurately approximate the true baselines. More sophisticated baseline estimation methods may be beneficial, and we leave their investigation to future work.

Table 8: Ablation studies with subtracting a baseline from the nonnegative $k3$ KL estimator. Experiments are done on Qwen3-1.7B-Base. Performances are measured by Pass@16 (%).

Method	MATH500	AIME24	AIME25	AMC23	OlympiadBench	MinervaMath
Forward $k3$ with baseline	75.21 (2.37)	20.68 (2.39)	10.45 (2.52)	**72.66** (2.95)	52.50 (0.60)	**46.86** (0.90)
Forward $k3$ without baseline	**75.62** (2.55)	**22.32** (3.22)	**18.94** (3.31)	70.55 (2.84)	**53.05** (0.66)	46.37 (0.98)
Reverse $k3$ with baseline	68.27 (0.75)	20.63 (2.55)	**19.89** (2.89)	76.36 (3.23)	54.14 (0.60)	47.82 (1.01)
Reverse $k3$ without baseline	**84.82** (3.43)	**32.23** (4.29)	11.76 (1.91)	**79.16** (3.83)	**58.57** (0.62)	**48.02** (1.02)

F.5 Ablation for Different Advantage Estimators

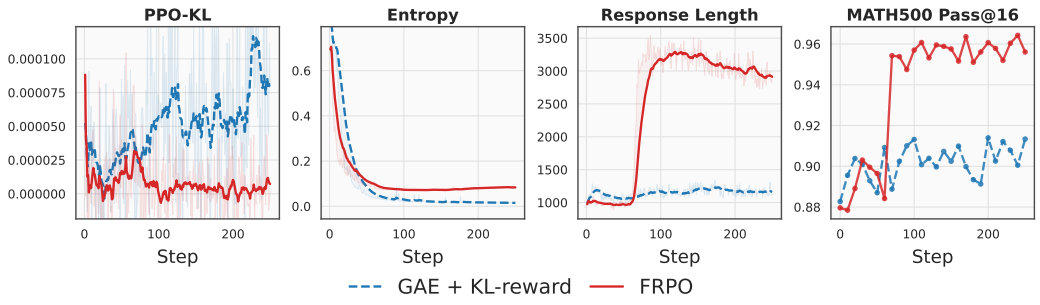


Figure 4: Comparison between GAE and GRPO advantage estimation, both with future KL.

In existing RL frameworks such as verl (Sheng et al., 2024) and AReaL (Fu et al., 2025), KL-in-reward can be combined with the GAE advantage estimator. This folds the regularizer into token rewards before advantage estimation. In contrast, FRPO preserves the GRPO-style group-normalized outcome advantage and adds future KL afterward. Figure 4, based on Qwen3-30B-A3B-Base, suggests that retaining GRPO's outcome-level normalization is beneficial compared with integrating KL into rewards before estimating advantages.

F.6 All-Equal Groups under KL-in-Reward

For $\beta > 0$, a group with equal verifier rewards receives the KL-in-reward advantage

$$\hat{A}_i^{\mathrm{KLR}} = -\frac{D_i - \bar{D}}{\mathrm{Std}(D) + \varepsilon_{\mathrm{std}}/\beta}, \qquad D_i = \sum_t \delta_{i,t}.$$

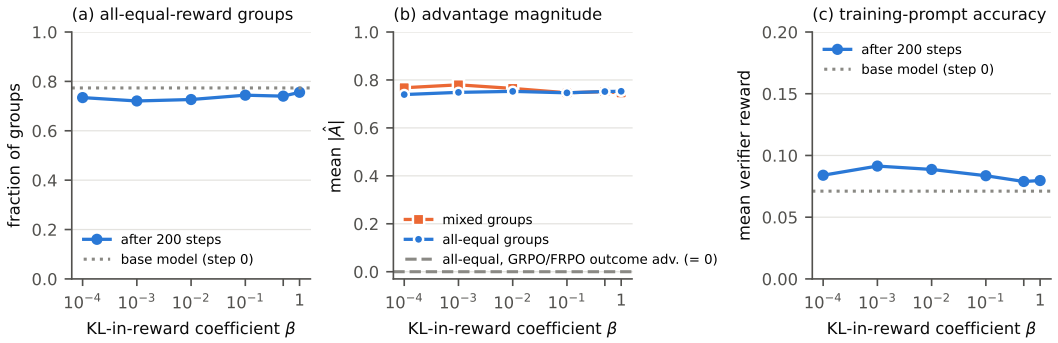


Figure 5: Replay of Qwen3-1.7B-Base and six step-200 KL-in-reward checkpoints on the same 512 training prompts ($k1$, $G = 5$; the unlabeled tick is $\beta = 0.5$). **(a)** All-equal-group fraction. **(b)** Mean advantage magnitude in all-equal and mixed groups; the dashed line marks zero outcome advantage under GRPO and FRPO. **(c)** Mean verifier reward.

When D_i varies and $\beta \operatorname{Std}(D) \gg \varepsilon_{\text{std}}$, normalization therefore produces an order-one signal nearly independent of β (Remark 13). Our implementation uses the sample standard deviation and $\varepsilon_{\text{std}} = 10^{-6}$, giving an RMS advantage approaching $\sqrt{(G-1)/G} \approx 0.89$ for $G = 5$.

We test six Qwen3-1.7B-Base runs with sampled $k1$, $\beta \in \{10^{-4}, 10^{-3}, 10^{-2}, 0.1, 0.5, 1\}$, $G = 5$, 256 prompts per step, no loss-side KL, and otherwise Table 5's settings. Without further training, we replay each step-200 checkpoint and the base model on the same 512 training prompts, using the training rollout settings, verifier, reward shaping, and advantage code. All-equal groups account for 77% of groups initially and 72–76% after training, almost all all-incorrect. Their mean advantage magnitude is 0.739–0.753, comparable to mixed groups (0.746–0.780), and they carry 75–79% of the token-weighted absolute advantage. Even at $\beta = 10^{-4}$, $\beta \operatorname{Std}(D)/\varepsilon_{\text{std}} \geq 24$ in every all-equal group. Similar magnitudes follow largely from normalization; the empirical findings are these groups' prevalence and share of the loss weights. Mean verifier reward remains 0.08–0.09, versus 0.07 initially. MATH500 mean@16, averaged over the last five evaluations, is 32.7–37.8%, versus 66.5% and 67.1% for single no-KL and loss-side-KL runs in the same setup.

Figure 6 applies the same replay protocol to one Qwen3-4B-Base run with $\beta = 10^{-5}$, the same group and batch sizes, and no loss-side KL. Here D_i sums verl's clipped $k3$ penalties: log-ratios are clipped to $[-20, 20]$ before applying Eq. (4), and token penalties are capped at 10. Between steps 20 and 160, all-equal groups increase from 56% to 77%, with mean advantage magnitude 0.69–0.75 and 55–76% of the token-weighted absolute advantage. Their advantage–length correlation is consistently negative (-0.12 to -0.19), favoring shorter sampled responses. Response length subsequently falls from about 800 tokens to fewer than 30 between steps 160 and 200. During collapse, token-identical groups receive zero advantage; excluding them leaves mean magnitude 0.60–0.68. This signal precedes collapse and is directionally consistent with it, but the single-run replay does not establish causation.

FRPO retains zero outcome advantage and adds β-scaled future-KL credit. Evaluating its default $k1$ credit on these same all-equal samples at $\beta = 10^{-5}$ gives mean magnitude below 3×10^{-4} over steps 20–160, including the implementation's $-\beta$ constant, versus 0.73–0.78 for KL-in-reward. This fixed-sample comparison illustrates the difference in scaling; our main FRPO runs use $\beta = 10^{-2}$ (Table 6).

F.7 Horizon Tradeoff in Future-KL Credit Weights

The full-horizon credit in Eq. (11) follows the on-policy gradient of Theorem 4. We study how truncating it changes its scalar magnitude, variability, and downstream performance. These credit statistics do not measure gradient variance or bias: the gradient also depends on the score vector, and an omitted tail affects its expectation only through the action-dependent component.

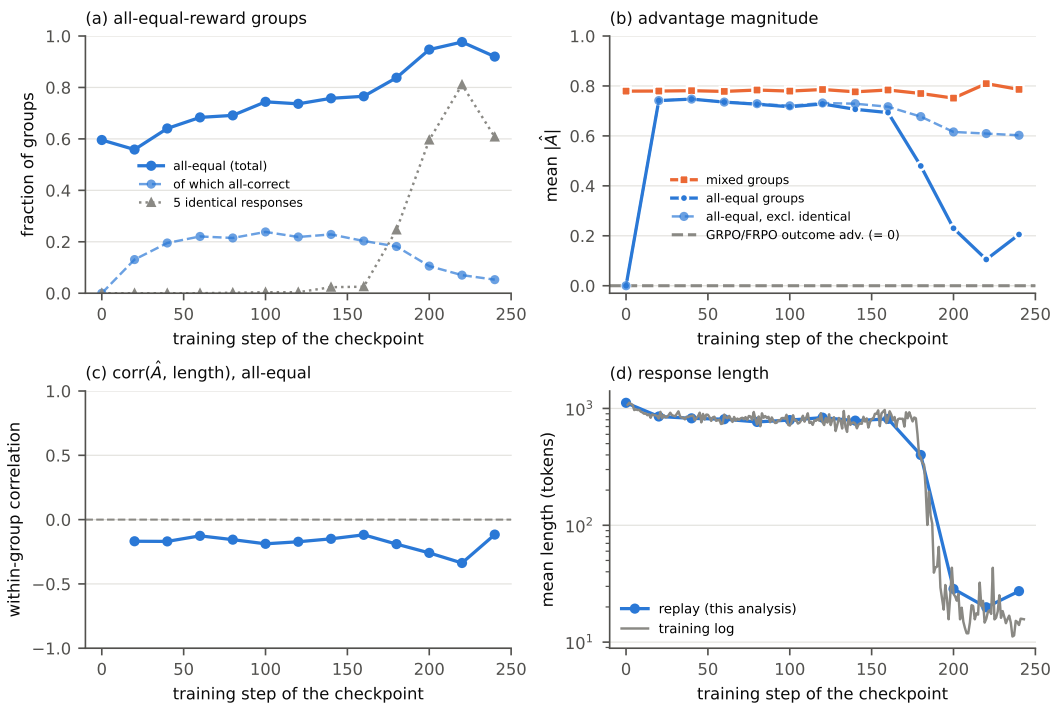


Figure 6: Replay every 20 steps of one Qwen3-4B-Base KL-in-reward run (clipped $k3$, $\beta = 10^{-5}$). **(a)** All-equal groups, including all-correct and token-identical groups. **(b)** Mean advantage magnitude, also excluding token-identical groups. **(c)** Mean within-group advantage–length correlation in all-equal groups. **(d)** Replay and training response lengths.

Setup. We compare seven Qwen3-1.7B-Base runs: FRPO with $W \in \{8, 64, 512, \infty\}$,[1] a whole-sequence variant, GRPO without KL, and GRPO with loss-side $k3$ KL (coefficient 10^{-3}). Training uses filtered DAPO-Math-17k and Table 6; the credit-based variants use reverse $k1$, $\beta = 0.01$, and recompute credit before each actor update (Appendix F.12). We compare the first 600 steps and report the continuation to 1,000 below.

For diagnostics, let $\delta_j = \log \pi_{\theta_{\text{old}}}(a_j \mid s_j) - \log \pi_{\text{ref}}(a_j \mid s_j)$, with $x_W(t)$ defined as in Eq. (11), $x_\infty(t) = \sum_{j \geq t} \delta_j$, and $x_{\text{seq}} = \sum_j \delta_j$. The whole-sequence variant assigns the latter to every token; its additional past terms are action-independent baselines for the ideal unclipped on-policy gradient. At every step, we log token-level moments and correlations for $W \in \{1, 2, 4, \ldots, 2048, \infty\}$ and x_{seq}. In particular, the token-averaged x_{seq} equals $\sum_i T_i D_i / \sum_i T_i$, where $D_i = \sum_j \delta_{i,j}$; this is a length-weighted diagnostic, not the ordinary trajectory-average KL. We emphasize MATH500 because each AIME set contains only 30 problems; the longer training also prevents direct comparison with the main-result table.

Credit distribution across horizons. We decompose the full credit into lag buckets,

$$x_\infty = \underbrace{x_8}_{B_1} + \underbrace{(x_{64} - x_8)}_{B_2} + \underbrace{(x_{512} - x_{64})}_{B_3} + \underbrace{(x_\infty - x_{512})}_{B_4},$$

covering tokens 1–8, 9–64, 65–512, and beyond 512, counting the current token first. Their mean shares $\mathbb{E}[B_k]/\mathbb{E}[x_\infty]$ and covariance shares $\text{Cov}(B_k, x_\infty)/\text{Var}(x_\infty)$ each sum to one. We recover bucket moments from the logged window moments and correlations, using ratios of step-averaged moments over steps 1–600; correlations are averaged across steps.

[1] The implementation counts future tokens after the current one: a run labeled W uses $x_{W+1} + 1$. The offset changes the statistics by about 10% at $W = 8$ and negligibly at $W \in \{64, 512\}$. The constant $+1$ has zero expected score-function gradient before clipping.

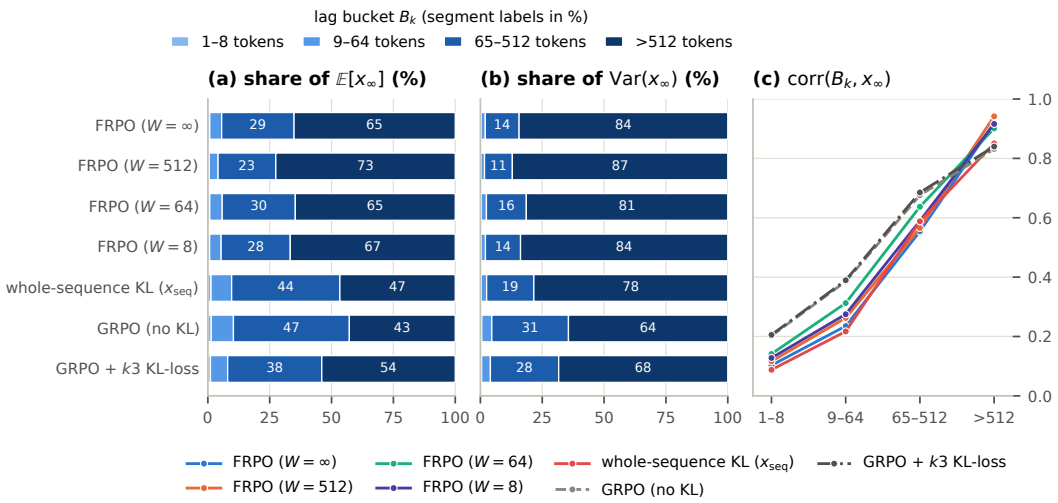


Figure 7: Lag-bucket decomposition over steps 1–600: **(a)** mean shares, **(b)** covariance shares of the full-credit variance (each row sums to 100%), and **(c)** token-level correlations with x_∞. In all seven runs, the tail beyond 512 tokens has a larger covariance share than mean share.

Table 9: Credit statistics on the default $W = \infty$ run over steps 1–600: ratios of step-averaged moments, mean token-level Pearson correlations, and $\beta\,\mathrm{SD}(x_W)$ for $\beta = 0.01$. MATH500 mean@16 (%) instead evaluates each separately trained horizon at step 600 and its best checkpoint up to that step; $W = 1$ was not trained.

	Credit statistics (default run)				**MATH500 mean@16**	
Horizon W	$\mathbb{E}[x_W]/\mathbb{E}[x_\infty]$	$\mathrm{Var}(x_W)/\mathrm{Var}(x_\infty)$	$\rho(x_W, x_\infty)$	$\beta\,\mathrm{SD}(x_W)$	step 600	best
1 (current token)	0.1%	0.014%	0.05	0.001	–	–
8	0.8%	0.092%	0.10	0.004	69.2	69.8
64	5.6%	0.77%	0.25	0.011	70.0	70.0
512	34.8%	8.8%	0.57	0.038	**70.3**	**70.3**
∞ (FRPO)	100%	100%	1	0.127	68.1	68.1
whole sequence (x_{seq})	3.35×	1.78×	0.59	0.170	61.2	63.6

In the default run, the tail beyond 512 tokens accounts for 65% of the mean credit and 84% of its covariance-based variance allocation, versus 29% and 14% for tokens 65–512 (Figure 7). Some variability reflects position and remaining length, since the tail vanishes on the last 512 tokens. On these same rollouts, $W = 512$ retains 35% of the mean at 9% of the variance, reducing $\beta\,\mathrm{SD}(x_W)$ from 0.127 to 0.038; the whole-sequence credit has 1.78 times the full-horizon variance (Table 9). These are empirical scalar-credit comparisons, not general variance-reduction guarantees or comparisons of gradient noise across trained runs.

Table 10: Downstream results at step 600. AIME24/25 mean@16 (%) averages evaluations at steps 560–600. MATH500 reports mean@16 at step 600 and the best checkpoint up to that step, plus exact pass@16 at step 600 (at least one correct response among 16 samples). Rollout statistics average steps 551–600; both KL credit means are token-weighted, in nats. Best accuracies are bold.

Method	**AIME24**	**AIME25**	**MATH500 mean@16**		**MATH500**	**Rollout statistics**			
	mean@16	mean@16	step 600	best	pass@16	Resp. len.	Entropy	$\mathbb{E}[x_\infty]$	$\mathbb{E}[x_{\mathrm{seq}}]$
GRPO (no KL)	9.3	4.3	67.4	67.7	86.0	1680	0.028	60.1	171.9
GRPO + $k3$ KL-loss (10^{-3})	9.2	5.5	66.4	69.1	85.4	2305	0.024	56.2	180.3
whole-sequence KL (x_{seq})	4.8	3.0	61.2	63.6	86.0	884	0.369	8.5	18.7
FRPO ($W = \infty$, default)	9.0	6.5	68.1	68.1	92.0	2884	0.117	11.8	41.6
FRPO ($W = 512$)	**10.0**	6.9	**70.3**	**70.3**	**93.2**	3917	0.092	16.1	54.9
FRPO ($W = 64$)	**10.0**	**7.1**	70.0	70.0	90.4	3382	0.084	21.3	71.8
FRPO ($W = 8$)	9.8	6.5	69.2	69.8	90.8	3243	0.090	25.2	78.6

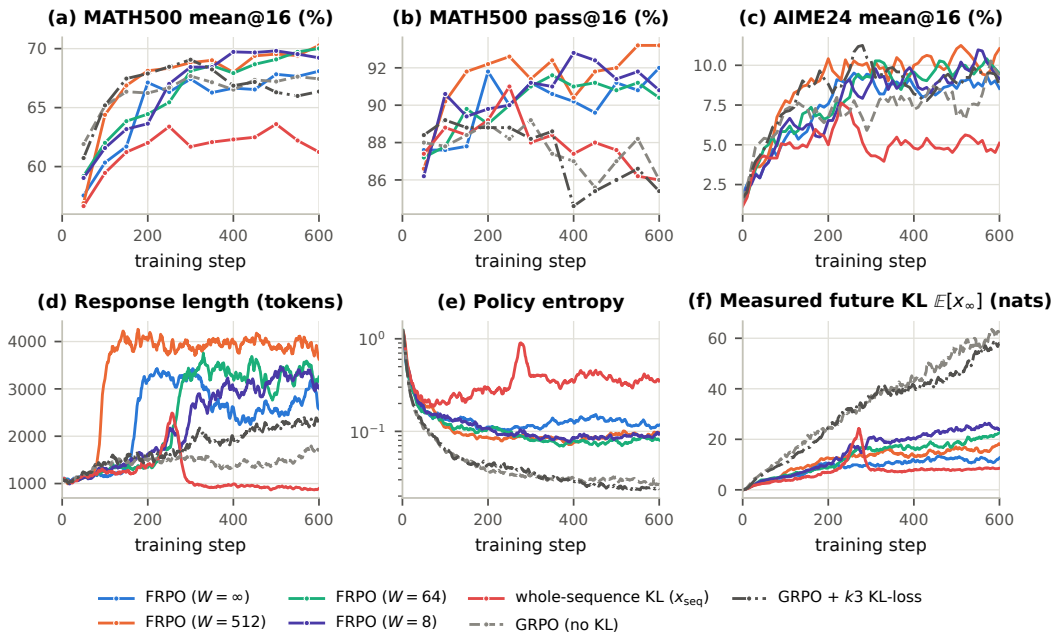


Figure 8: Training dynamics: **(a,b)** offline MATH500 mean@16 and pass@16, **(c)** AIME24 mean@16 (three-evaluation rolling mean), **(d)** response length, **(e)** policy entropy, and **(f)** token-averaged x_∞. Panels (d)–(f) use centered five-step rolling means.

Downstream results and interpretation. At step 600, all FRPO horizons exceed both GRPO baselines on MATH500 (Table 10). $W = 512$ attains the highest mean@16/pass@16 ($70.3\%/93.2\%$), followed by the full horizon on pass@16 (92.0%). The credit-based variants retain higher entropy and lower token-weighted KL statistics than the baselines. The whole-sequence variant performs worst among them, with mean@16 of 61.2%; its responses shorten from about 2,500 to 900 tokens after step 260, while FRPO retains roughly 2,900–3,900 tokens over steps 551–600 (Figure 8). At fixed β, truncation also removes much of the cost-to-go; the windowed runs have higher token-weighted KL and longer responses than the full-horizon run, consistent with weaker effective regularization.

F.8 Sensitivity to the Regularization Coefficient

Setup. We sweep the regularization coefficient of FRPO (full-horizon reverse $k1$ future KL, $W = \infty$, recomputed under the current policy before each actor update as in Appendix F.12) and of GRPO with the loss-side $k3$ penalty, whose coefficient we set to the same β, over $\beta \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$. Each run trains Qwen3-1.7B-Base for 500 steps with the hyperparameters of Tables 5 and 6 other than β, on the filtered DAPO-Math-17k training set. MATH500 is evaluated during training with 16 samples per problem at temperature 1.0 and an 8,192-token budget. We report mean@16 and pass@16, the fraction of problems solved by at least one of the 16 samples, together with the measured KL to the reference $\mathbb{E}[x_{\text{seq}}]$ (the token-averaged sampled $k1$ KL of the whole response, as in Appendix F.7), the response length, and the policy entropy.

Results. $\beta = 10^{-2}$ gives the highest mean@16 and pass@16 of FRPO in the sweep, while for loss-side KL, $\beta = 10^{-3}$ achieves the best pass@16 performance, which is the main metric we care. Besides, the future-KL credit does not lengthen responses. At every β, FRPO generates no longer responses than the loss-side penalty. The longer responses of FRPO in the main comparisons therefore come from the regularization strength it uses, which preserves exploration, and not from a length bias of the future-KL credit.

Therefore, we keep $\beta = 10^{-2}$ for FRPO and $\beta = 10^{-3}$, which is also the default configuration in the original VeRL codebase, for the loss-side KL.

Table 11: Sensitivity to the regularization coefficient on Qwen3-1.7B-Base after 500 training steps. MATH500 mean@16 and pass@16 (%) are reported, with the measured KL divergence to the reference $\mathbb{E}[x_{\text{seq}}]$ (nats, token-averaged), response length (tokens), and entropy.

		MATH500		Rollout statistics		
Method	β	mean@16	pass@16	$\mathbb{E}[x_{\text{seq}}]$	Resp. len.	Entropy
FRPO	10^{-4}	64.5	83.2	108.7	1144	0.032
FRPO	10^{-3}	66.8	85.6	88.6	1256	0.034
FRPO	10^{-2}	67.13	91.8	39.9	2575	0.136
FRPO	10^{-1}	48.7	85.6	2.3	963	0.582
GRPO + loss-side $k3$	10^{-4}	64.4	83.6	136.3	1526	0.029
GRPO + loss-side $k3$	10^{-3}	65.9	86.3	106.6	1281	0.032
GRPO + loss-side $k3$	10^{-2}	69.89	84.78	43.3	3006	0.123
GRPO + loss-side $k3$	10^{-1}	47.1	84.7	2.8	1114	1.028

F.9 Experiments on Qwen3-4B-Base

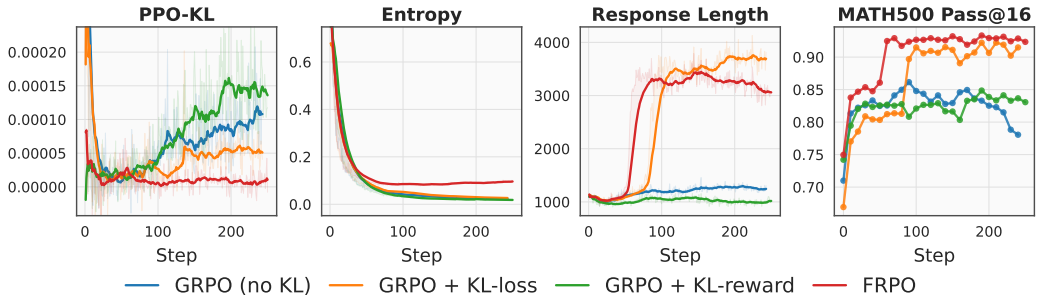


Figure 9: Training dynamics for Qwen3-4B-Base model with different KL integrations.

Figure 9 shows the training dynamics of Qwen3-4B-Base, trained with the same setup and hyperparameters as Qwen3-1.7B-Base (Tables 5 and 6). As in the 30B setting, FRPO maintains a higher entropy and a lower PPO-KL than the baselines, and it attains the highest MATH500 pass@16 over most of training.

F.10 Other Evaluation Metrics

In previous sections, we mainly focused on the metric pass@n, as we hope that KL regularization could help maintain the base model's inherent ability and alleviate entropy collapse. From Figure 2 and Figure 9, we could see that the convergence entropy of FRPO is indeed at a higher value, leading to a higher pass@n performance. Here, we also show the mean@n performance.

From Figure 10, Figure 11, and Figure 12, we can see that FRPO not only improves pass@n performance but also achieves a comparable or even better mean@n accuracy than other methods that directly put KL in the loss or the reward. Especially on Qwen3-30B-A3B-Base, where off-policy issues are more pronounced than in other dense models due to its MoE structure, using future KL in FRPO helps stabilize training and yields a smoother training process.

F.11 Full Evaluation Results

We use AIME24, AIME25, AMC23, OlympiadBench (He et al., 2024), and MinervaMath (Lewkowycz et al., 2022) for evaluation besides MATH500. The information about each benchmark can be found in Table 12

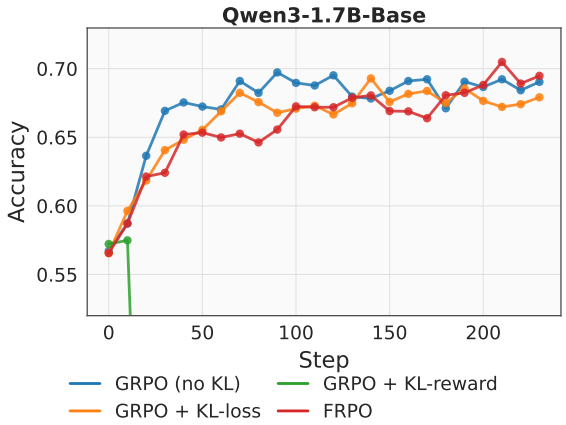


Figure 10: The training dynamics of mean@16 accuracy on MATH500 for Qwen3-1.7B-Base under different KL integrations.

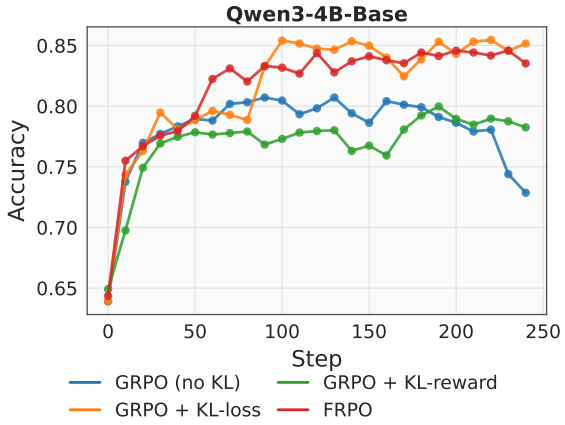


Figure 11: The training dynamics of mean@16 accuracy on MATH500 for Qwen3-4B-Base under different KL integrations.

Table 1 and Table 13 summarize the evaluation results of all models with different KL integrations on several benchmarks for pass@16 and mean@16, respectively. Note that using (reverse) KL in the reward directly leads to model collapse.

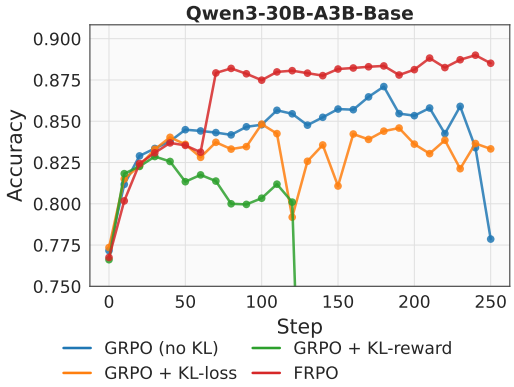


Figure 12: The training dynamics of mean@16 accuracy on MATH500 for Qwen3-30B-A3B-Base under different KL integrations.

Table 12: Information about benchmarks.

Benchmark	**Size**
AIME24	30
AIME25	30
AMC23	40
OlympiadBench	674
MinervaMath	272

Table 13: Full evaluation results of mean@16 (%) for Qwen3-1.7B-Base and Qwen3-30B-A3B-Base on different benchmarks.

Model	**MATH500**	**AIME24**	**AIME25**	**AMC23**	**OlympiadBench**	**MinervaMath**
Qwen3-1.7B-Base	56.71	2.08	0.83	13.44	11.21	9.79
+ GRPO (no KL)	69.23	8.33	4.17	40.16	**33.25**	**28.33**
+ GRPO KL-loss	69.29	9.38	4.58	**42.03**	30.80	27.23
+ GRPO KL-reward	/	/	/	/	/	/
+ FRPO	**70.90**	**9.79**	**4.79**	41.56	31.64	25.46
Qwen3-30B-A3B-Base	77.16	9.17	3.96	33.28	23.44	18.45
+ GRPO (no KL)	87.10	24.17	17.71	75.16	51.00	**44.28**
+ GRPO KL-loss	84.83	22.29	10.21	66.56	47.53	42.69
+ GRPO KL-reward	/	/	/	/	/	/
+ FRPO	**88.13**	**32.08**	**24.17**	**75.63**	**58.32**	43.18

F.12 MINI-BATCH STALENESS, PPO CLIPPING, AND ON-POLICY TRAINING

Setup. To fully utilize the computation resources for the rollout engine, the rollout train batch is larger than the actor update mini-batch. In our main experiments, the train batch size is 256 prompts and the PPO mini-batch size is 64 prompts. Thus, a rollout stage samples responses for 256 prompts at once from $\pi_{\theta_{\text{old}}}$, and the actor then takes one optimizer step on each of four mini-batches. The PPO epoch is set to 1, so each mini-batch is used once rather than repeatedly optimized, and the future-KL term is computed once per rollout batch from the log-probabilities at θ_{old} (Algorithm 1). Write θ_k for the parameters before the update on mini-batch $k \in \{0, 1, 2, 3\}$. Mini-batch 0 has no staleness ($\theta_0 = \theta_{\text{old}}$, so $\rho \equiv 1$ and clipping is inactive), while mini-batches 1–3 are updated at $\theta_k \neq \theta_{\text{old}}$ with the rollout-time advantage, including its future-KL term. Proposition 17 bounds the population surrogate gradient error by $O(\|\theta_k - \theta_{\text{old}}\|)$ while clipping is inactive; finite-batch sampling and normalization are separate effects. Lemma 16 shows that clipping gates the combined reward and future-KL weight. Here we measure both effects on Qwen3-1.7B-Base with the configuration of Table 6 (reverse KL, $k1$, $\beta = 0.01$, 200 steps). Accuracy is MATH500 pass@16 and mean@16 averaged over the last five in-training evaluations.

Table 14: Per-mini-batch staleness and clipping statistics of the default FRPO run on Qwen3-1.7B-Base, aggregated over all 200 training steps (about 5.4×10^8 response-token updates). Token share is the fraction of response tokens in each mini-batch; the shares differ because the trainer balances sequence lengths across data-parallel ranks before the split. Gated tokens are those whose gradient the clipped objective sets exactly to zero, split by the sign of the FRPO advantage (as a percentage of all tokens). Cosine and relative ℓ_2 compare the realized per-token loss coefficients $c_t = \rho_t \hat{A}_t \mathbf{1}[t \text{ not gated}]$ with their unclipped counterparts $\rho_t \hat{A}_t$, with relative $\ell_2 = \|c - c^{\text{unclip}}\|/\|c\|$.

Mini-batch	Token share	Mean ρ	Std ρ	Gated (%)	Gated, $\hat{A} > 0$ (%)	Gated, $\hat{A} < 0$ (%)	Cosine	Rel. ℓ_2
0	10.0%	1	0	0	0	0	1	0
1	38.8%	1.0004	0.050	0.176	0.038	0.138	0.9985	0.055
2	40.9%	1.0004	0.031	0.173	0.036	0.138	0.9986	0.054
3	10.3%	1.0006	0.036	0.324	0.118	0.206	0.9974	0.072
All	100%	1.0004	0.039	0.173	0.041	0.131	0.9986	0.054

How far the policy drifts within a rollout batch. The actor logs, for every mini-batch, the ratio ρ_t, whether the clipped objective zeroes the token's gradient, and the per-token loss coefficient with and without clipping. Table 14 aggregates these statistics for the default run. Mini-batch 0 has $\rho_t = 1$ exactly and no gated token. In mini-batches 1–3, 0.17–0.32% of the token updates are gated, and the realized coefficient vector has cosine at least 0.997 with its unclipped counterpart. The pooled standard deviation of ρ_t is 0.03–0.05, and its median over steps is 0.030. Two additional seeds of the default configuration give pooled gated fractions of 0.18% and 0.19% and cosines of 0.9980 and 0.9984. The cosine and relative ℓ_2 distance describe the token weights that multiply $\nabla_\theta \log \pi_\theta(a_t \mid s_t)$, rather than the parameter gradient itself.

The stale and recomputed future-KL credits differ by $\beta \sum_{j \geq t} \log \rho_j$ (Appendix C.5). Under sampling from $\pi_{\theta_{\text{old}}}$, the future terms $j > t$ have conditional mean $-D_{\text{KL}}(\pi_{\theta_{\text{old}}}(\cdot \mid s_j) \| \pi_{\theta_k}(\cdot \mid s_j))$, which is of second order in $\theta_k - \theta_{\text{old}}$, and the difference also has a zero-mean fluctuation of first order. Taking the median standard deviation of ρ_t (0.030) as a proxy for that of $\log \rho_t$ and treating the per-token terms as uncorrelated, this fluctuation is about $\beta \cdot 0.03 \cdot \sqrt{3{,}000} \approx 0.016$ over a 3,000-token suffix, compared with a root mean square of 0.49 for the reward advantage and 0.11 for the future-KL credit in this run. The recomputation ablation below measures its effect on training.

No detectable effect of staleness or clipping at this ratio. Within the first group of Table 15, recomputing the future-KL term under the current policy π_{θ_k} before every mini-batch update changes pass@16 by -0.49 and mean@16 by $+0.17$ points. This ablation targets the stale-log-ratio part of the error in Proposition 17: it approximately restores the gradient of the proximal term and leaves the prefix-distribution and gain terms unchanged. Removing clipping altogether changes pass@16 and mean@16 by -1.42 and -0.89 points. Both ablations are single runs, and both changes are smaller than the spread of the default configuration across its three seeds (2.05 points in pass@16 and 2.40 points in mean@16), so at four mini-batches per rollout batch we detect no effect of the stale future-KL term or of clipping on accuracy.

Table 15: Effect of the staleness of the future-KL term, of PPO clipping, and of on-policy training on Qwen3-1.7B-Base. MATH500 pass@16 and mean@16 (%) are averaged over the last five in-training evaluations. The comparison with on-policy training (second group) holds the loss aggregation fixed and matches the number of parameter updates.

Configuration	Prompts / rollout	Updates / rollout	Total updates	pass@16	mean@16
Token-mean loss (default)					
FRPO, future KL fixed at θ_{old}	256	4	800	90.13	66.86
FRPO, future KL fixed at θ_{old}	256	4	800	89.34	66.03
future KL recomputed under π_{θ_k} at every mini-batch	256	4	800	89.64	67.03
no clipping ($\epsilon = \bar{c} = 10^6$)	256	4	800	88.71	65.97
Sequence-mean loss					
FRPO, four mini-batches per rollout batch	256	4	800	89.39	67.11
FRPO, on-policy (one mini-batch per rollout batch)	64	1	800	89.37	65.33

Comparison with on-policy training. On-policy training uses a single mini-batch per rollout batch, so every update is taken at θ_{old} with $\rho \equiv 1$, which we verified in the logs. To isolate the effect of reusing a rollout batch for several updates, we compare the two schedules at a matched number of parameter updates and with the same loss aggregation. We use a sequence-mean loss for this comparison, because its weighting of responses does not depend on how the rollout batch is partitioned into mini-batches. The on-policy run samples 64 prompts per rollout, which also matches the per-update batch, the total number of generated responses, and the number of passes over the data. Four sequential mini-batch updates per rollout batch and strictly on-policy training reach 89.39 and 89.37 pass@16 and 67.11 and 65.33 mean@16 (single runs). Reusing each rollout batch for four updates therefore does not reduce accuracy at this ratio, while it requires a quarter as many rollout stages.

Summary. How large is the mismatch between the on-policy derivation and the rollout-based implementation? At this ratio it is small: the first mini-batch has $\rho \equiv 1$, later mini-batches have a median per-step standard deviation of ρ_t of 0.03, recomputing the future-KL term changes accuracy by less than the seed-to-seed spread, and with the same loss aggregation four updates per rollout batch match strictly on-policy training. Does PPO clipping preserve the intended future-KL gradient? On the 99.8% of token updates that are not gated, the clipped objective applies exactly the unclipped weight $\rho_t \hat{A}_t$, so the verifier and future-KL parts keep their derived ratio, and it removes the gradient of the 0.17% of token updates that are gated. At given ratios, the future-KL term affects clipping only through the sign of the advantage, which it determines in groups with identical rewards, where the GRPO advantage vanishes. Removing clipping altogether changes pass@16 by -1.4 points, within the seed-to-seed spread.