ESTIMATION DEBIASING FOR ASYNCHRONOUS POLICY OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

In reinforcement learning for large language models, asynchronous pipelines sample a batch of rollouts once and use them for several updates. This reduces generation overhead but degrades performance due to rollout staleness: the behavior policy gradually diverges from the target policy, introducing off-policy errors that bias the estimated objective. Existing methods control this error with PPO-style clipping by limiting the policy shift. However, our derivation shows that this restriction also limits objective improvement, making clipping alone insufficient. In addition to clipping, the bias should be directly corrected in the objective estimator. We propose Bias-Corrected Yielded Policy Optimization (BYPO), which constructs a more accurate objective estimator through sequence-level importance weighting and second-order Taylor correction, adaptively switches between them to control variance, and falls back to a first-order approximation when both corrections become excessively large. Experiments on mathematical reasoning and preference-alignment tasks show consistent improvements in asynchronous training. Applied to DAPO, BYPO mitigates 83.70% of the performance degradation caused by rollout staleness, with negligible runtime overhead.

1 INTRODUCTION

Reinforcement learning (RL) has become one of the dominant paradigms for post-training large language models (LLMs) (Guo et al., 2025; Achiam et al., 2023; Team et al., 2025). In this paradigm, the behavior policy first generates responses to each prompt, and these responses, together with their rewards, are used to estimate the expected reward as the optimization objective (Wang & Sun, 2025; Chaudhari et al., 2025). When the behavior policy is fully in sync with the policy being updated, the estimated objective is unbiased (Schulman et al., 2015). However, a fully synchronous pipeline requires fresh rollouts before every update, severely underutilizing GPUs and making training prohibitively slow (Hu et al., 2026; Tian et al., 2026). To improve efficiency, the most recent RL frameworks (Sheng et al., 2025; von Werra et al., 2020; Liang et al., 2025) instead adopt an asynchronous pipeline, where multiple batches of responses are generated at once and used across several iterations of parameter updates. This accelerates training, but introduces rollout staleness: the behavior policy drifts from the policy being optimized, biasing the estimated objective with an off-policy error (Shen et al., 2026; Roux et al., 2025). These off-policy errors arise from sequence-level policy mismatch and therefore cannot be fully corrected by PPO-style token-level importance weights (Schulman et al., 2017; Shao et al., 2024; Zheng et al., 2025a). Moreover, although constraining policy shift can indirectly control off-policy error (Schulman et al., 2015), our derivation shows that it also limits objective improvement of the same order, motivating stronger error-control mechanisms.

Fortunately, the structure of RL for LLMs provides opportunities to estimate the objective more accurately. First, the token-level response distributions are tractable, enabling an unbiased objective estimator via sequence-level importance weighting. However, this estimator suffers from prohibitively large variance; our analysis shows that the variance grows approximately as a high-degree polynomial in the policy shift. Although applying nonlinear transformations to the importance weights (Shen et al., 2026; Roux et al., 2025) can reduce variance, these transformations introduce additional bias and have not been shown to preserve the benefits of importance weighting. This motivates a lower-variance estimator. Our derivation shows that existing methods such as DAPO (Yu et al., 2026) can be interpreted as optimizing a first-order Taylor approximation of

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.

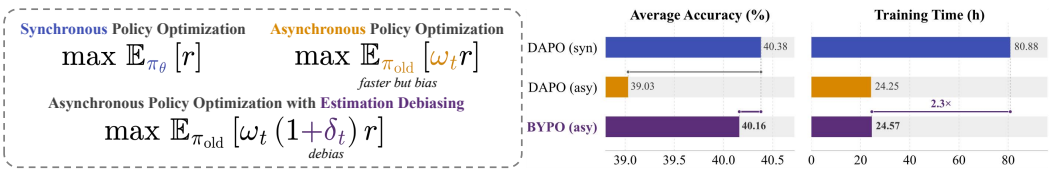


Figure 1: BYPO debiases the off-policy error in asynchronous policy optimization caused by rollout staleness through the correction term δ_t (Left). This correction mitigates most of the performance degradation caused by rollout staleness while preserving the inference acceleration of the asynchronous paradigm (Right).

the true objective around the behavior policy. Accordingly, a second-order Taylor approximation provides a more accurate estimator. We find that it can be computed without resampling, owing to the acyclic structure of the autoregressive process. We further prove that this second-order approximation yields a lower-order off-policy error, while its variance scales only quadratically with the policy shift. Together, these properties make second-order correction particularly practical under large policy shifts.

Building on these observations, we develop a plug-and-play policy optimization method with estimation debiasing, called Bias-Corrected Yielded Policy Optimization (BYPO), which can be integrated into most existing RL algorithms for LLMs. When the sequence-level importance-sampling correction is small, BYPO applies it directly to obtain a more accurate estimate. When the correction becomes large, BYPO switches to a lower-variance second-order Taylor correction. If both correction terms become excessively large, BYPO disables them and falls back to DAPO's low-variance first-order approximation. Our analysis further shows that this adaptive switching controls variance effectively without introducing substantial additional bias.

To evaluate the effectiveness of BYPO, we conduct extensive experiments. We first train LLMs of different types and scales on multiple datasets for reasoning enhancement, and evaluate them on 10 standard mathematical reasoning benchmarks, where BYPO achieves the highest overall score. We observe that the correction term is activated in more than 85% of responses, highlighting its important role. As shown in Figure 1, BYPO reduces the performance degradation on benchmarks relative to synchronous training to 16.30%, demonstrating effective off-policy correction. Results on preference-alignment tasks further demonstrate BYPO's generality across RL-for-LLM settings.

In summary, the contributions of this work are as follows:

- We quantify the impact of off-policy error on policy optimization under rollout staleness and analyze the limitations of policy-shift constraints.
- We propose BYPO, which uses sequence-level importance weights and second-order Taylor correction to correct off-policy errors.
- We apply BYPO to both RLVR and RLHF settings, demonstrating broad and consistent improvements.

2 Related Work

Asynchronous RL for LLMs. RL trains LLMs through an exploration–exploitation process, with each iteration comprising rollout generation, reward computation, and policy updates (Wang & Sun, 2025). Collecting fresh rollouts before every update yields unbiased objective estimates but often results in low device utilization and prohibitively long training times (Hu et al., 2026; Tian et al., 2026). To improve efficiency, most recent RL frameworks decouple generation from optimization (Sheng et al., 2025; von Werra et al., 2020; Liang et al., 2025), generating a large batch of responses once and using them for multiple policy updates. This asynchronous approach reduces training time but introduces rollout staleness, where the mismatch between behavior and target policies causes an off-policy error between the estimated and true objectives (Schulman et al., 2015).

Off-policy error control. TRPO (Schulman et al., 2015) was among the first to analyze policy-mismatch-induced off-policy error and bound it using the divergence between the behavior and

target policy. Motivated by this analysis, PPO (Schulman et al., 2017) constrains policy updates through likelihood-ratio clipping, which indirectly mitigates off-policy error. Recent work on asynchronous RL for LLMs has also explored using full-sequence importance weights to correct policy mismatch (Shen et al., 2026; Roux et al., 2025), while applying smooth nonlinear transformations to control the resulting variance. However, these transformations alter the importance weights and introduce additional bias, whose effect on the corrective benefit remains theoretically unclear. In contrast, BYPO adaptively switches among correction schemes with different variance characteristics, maintaining stability without reshaping the correction terms.

3 PRELIMINARY

RL for LLMs. Most RL methods for LLMs adopt a token-level formulation. In step t, the prompt s together with the generated prefix $a_{0:t} = a_{<t}$ defines the conditional state. The next token a_t is then sampled from the LLM policy $\pi_\theta(\cdot|a_{0:t}, s)$. Each complete response $a_{0:T}$ receives a reward r, representing either a preference-alignment score in RLHF or a correctness score in RLVR. The goal of RL for LLMs is to optimize the model policy by maximizing the expected reward as:

$$\mathcal{L}(\pi_\theta) = \mathbb{E}_{p(a_{0:T}, s|\pi_\theta)}\left[r\right], \tag{1}$$

where we pad all responses to length T for simplicity. This padding does not affect the subsequent derivations, since the added variables can be treated as constants.

Off-Policy Error in Asynchronous Token-Level RL. Although Equation 1 can be optimized unbiasedly with online Monte Carlo samples (Williams, 1992), this requires sampling directly from π_θ, i.e., collecting new trajectories before each parameter update. Such a synchronous pipeline severely limits GPU utilization and makes training time prohibitively long (Hu et al., 2026; Tian et al., 2026). Therefore, most implementations estimate the objective using rollouts generated by a stale policy π_{old} and effectively optimize the following surrogate objective (Schulman et al., 2015; 2017; Shao et al., 2024; Zheng et al., 2025a):

$$L_{\pi_{\text{old}}}(\pi_\theta) = \mathbb{E}_{p(a_{0:T}, s|\pi_{\text{old}})}\left[\sum_t k_t \omega_t r\right] + \zeta(\pi_{\text{old}}), \tag{2}$$

where π_{old} is the behavior policy, $\omega_t = \frac{\pi_\theta(a_t|a_{0:t}, s)}{\pi_{\text{old}}(a_t|a_{0:t}, s)}$ is the token-level importance weight, and k_t is the aggregation weight for the t-th token-level loss. The weight k_t is constant in algorithms such as DAPO (Yu et al., 2026) and Dr-GRPO (Liu et al., 2025), but inversely proportional to the response length in GRPO (Shao et al., 2024). And $\zeta(\pi_{\text{old}}) = \mathcal{L}(\pi_{\text{old}}) - \mathbb{E}_{\pi_{\text{old}}}\left[\sum_t k_t r\right]$ is constant with respect to π_θ and therefore need not be explicitly included in the loss.

When $k_t \equiv 1$ and training is fully synchronous ($\pi_\theta = \pi_{\text{old}}$), the surrogate objective $L_{\pi_{\text{old}}}(\pi_\theta)$ matches the true objective $\mathcal{L}(\pi_\theta)$ in both value and gradient (Schulman et al., 2015). However, this equivalence generally breaks when $\pi_\theta \neq \pi_{\text{old}}$, giving rise to the off-policy error

$$\epsilon_{\text{off}} = \left|\mathcal{L}(\pi_\theta) - L_{\pi_{\text{old}}}(\pi_\theta)\right|, \tag{3}$$

which measures the gap between the true objective and the surrogate objective optimized in asynchronous RL. Note that existing algorithms typically replace the raw reward r with an advantage function $A^{\pi_{\text{old}}} \propto r - \mathbb{E}_{\pi_{\text{old}}}[r]$ to reduce variance without changing the policy gradient. This is equivalent to replacing $L_{\pi_{\text{old}}}(\pi_\theta)$ with $L_{\pi_{\text{old}}}(\pi_\theta) - L_{\pi_{\text{old}}}(\pi_{\text{old}})$, which differs only by a constant. Therefore, our analysis applies directly to advantage-based methods.

Policy-Shift Constraints for Off-Policy Error Control. Although off-policy error exists, its magnitude can be controlled. According to prior results, when $k_t \equiv 1$ and rewards are bounded, the error admits the following upper bound (Han et al., 2026):

$$\epsilon_{\text{off}} \leq \tau\, \mathbb{E}_{\pi_{\text{old}}}\left[\sum_t \mathbb{D}_{\text{TV}}\left(\pi_\theta(\cdot|a_{0:t}, s) \,\|\, \pi_{\text{old}}(\cdot|a_{0:t}, s)\right)\right], \tag{4}$$

where $\mathbb{D}_{\text{TV}}$ denotes the total-variation (TV) divergence and τ is a constant. Equation 4 shows that constraining the shift between π_θ and π_{old} bounds the off-policy error and can thereby help ensure improvement in the true objective in Equation 1. This explains the widespread use of clipping mechanisms in policy optimization algorithms (Schulman et al., 2017; Shao et al., 2024; Yu et al., 2026). However, we later show that the same TV divergence also upper-bounds the improvement of the surrogate objective.

4 POLICY-SHIFT CONSTRAINTS ARE INSUFFICIENT

Although the off-policy error ϵ_{off} is upper-bounded by the TV divergence in Equation 4, the improvement of the surrogate objective $L_{\pi_{\text{old}}}(\pi_\theta)$ in Equation 2 is also bounded at the same order. Since Equation 2 is a linear function of π_θ and the TV divergence is piecewise linear in π_θ, the maximum improvement is achieved by shifting probability mass from the lowest-advantage actions to the highest-advantage actions. Therefore, we have:

$$\begin{aligned} L_{\pi_{\text{old}}}(\pi_\theta) - L_{\pi_{\text{old}}}(\pi_{\text{old}}) &= \mathbb{E}_{\pi_{\text{old}}}\left[\sum_t k_t \left(\mathbb{E}_{\pi_\theta}\left[Q_t^{\pi_{\text{old}}} | a_{0:t}, s\right] - \mathbb{E}_{\pi_{\text{old}}}\left[Q_t^{\pi_{\text{old}}} | a_{0:t}, s\right]\right)\right] \\ &\leq Q^{\Delta} \mathbb{E}_{\pi_{\text{old}}}\left[\sum_t \mathbb{D}_{\text{TV}}\left(\pi_\theta(\cdot|a_{0:t}, s) \,\|\, \pi_{\text{old}}(\cdot|a_{0:t}, s)\right)\right], \end{aligned} \tag{5}$$

where $Q_t^{\pi_{\text{old}}} = \mathbb{E}_{\pi_{\text{old}}}[r|a_{0:t+1}, s]$ has the same meaning as the action-value function in RL, and $Q^{\Delta} = \max k_t Q_t^{\pi_{\text{old}}} - \min k_t Q_t^{\pi_{\text{old}}}$. Equation 5 shows that the per-iteration improvement in the surrogate objective is also upper-bounded by the TV divergence. Consequently, the upper bound on the estimated objective improvement and the upper bound on the corresponding estimation error are of the same order. This means that even when the estimated objective improvement is substantial, the true objective may improve only marginally, resulting in ineffective policy updates. To further improve the efficiency of RL for LLMs, we need a more accurate objective estimator for asynchronous policy optimization that directly reduces the error in the estimated objective improvement.

5 METHODOLOGY

5.1 SEQUENCE-LEVEL IMPORTANCE WEIGHTS CORRECTION

Correction Formulation. Since the state and action distributions at each step are estimable in RL for LLMs, we can unbiasedly estimate the true expected reward $\mathcal{L}(\pi_\theta)$ off-policy using sequence-level importance weights:

$$\eta_{\text{is}} = r \, \frac{p(a_{0:T}|s; \pi_\theta)}{p(a_{0:T}|s; \pi_{\text{old}})} = r \prod_t \omega_t, \tag{6}$$

where $\prod_t \omega_t = \prod_t \frac{\pi_\theta(a_t|a_{0:t}, s)}{\pi_{\text{old}}(a_t|a_{0:t}, s)}$ is the sequence-level importance weight.

Unbiasedness. The expectation of η_{is} in Equation 6 is equal to the true expected reward $\mathcal{L}(\pi_\theta)$ in Equation 1:

$$\mathbb{E}_{\pi_{\text{old}}}[\eta_{\text{is}}] = \mathbb{E}_{p(s)}\left[\mathbb{E}_{p(a_{0:T}|s;\pi_{\text{old}})}\left[\frac{p(a_{0:T}|s;\pi_\theta)}{p(a_{0:T}|s;\pi_{\text{old}})} r\right]\right] = \mathbb{E}_{p(s)}\left[\mathbb{E}_{p(a_{0:T}|s;\pi_\theta)}[r]\right] = \mathcal{L}(\pi_\theta), \tag{7}$$

therefore is unbiased.

Variance. Despite being unbiased, sequence-level importance sampling can exhibit prohibitively large variance. Assume that the importance weight ω_t has a finite second moment and that f is a random variable with a finite fourth moment satisfying $\mathbb{E}_{\pi_\theta(a_t|s,a_{0:t})}[f^2] > 0$. We then have

$$\mathbb{E}_{\pi_{\text{old}}(a_t|s,a_{0:t})}[\omega_t^2 f^2] = C_t(1 + \mathbb{D}_t)\mathbb{E}_{\pi_\theta(a_t|s,a_{0:t})}[f^2], \tag{8}$$

where $\mathbb{D}_t = \chi^2(\pi_\theta(\cdot \mid a_{0:t}, s) \,\|\, \pi_{\text{old}}(\cdot \mid a_{0:t}, s))$ denotes the token-level chi-square divergence between π_θ and π_{old} at position t, and $C_t = 1 + \frac{\text{Cov}_{\pi_\theta(a_t|s,a_{0:t})}(\omega_t, f^2)}{\mathbb{E}_{\pi_\theta(a_t|s,a_{0:t})}[\omega_t]\,\mathbb{E}_{\pi_\theta(a_t|s,a_{0:t})}[f^2]}$ can be readily shown to be bounded and positive under these assumptions. Substituting these expressions into the law of total variance shows that there exists a bounded constant K_t such that

$$\text{Var}_{\pi_{\text{old}}}[(\omega_t - 1)f] = O\left(\mathbb{E}_{\pi_{\text{old}}}\left[1 + K_t \, \mathbb{E}_{\pi_{\text{old}}}\left[f^2\right] \mathbb{D}_t\right]\right). \tag{9}$$

Note that Equation 9 should be viewed only as a heuristic, order-of-magnitude characterization for comparing the variances of different correction terms, rather than as a rigorous variance bound.

By setting $f = r \prod_{t>i} \omega_t$ recursively and expanding over i using Equation 9, we obtain the variance scaling of the importance correction η_{is}:

$$\text{Var}_{\pi_{\text{old}}}[\eta_{\text{is}}] = O\left(\mathbb{E}_{\pi_{\text{old}}}\left[\prod_i (1 + K_i \mathbb{D}_i)\right]\right), \tag{10}$$

where the variance scales polynomially with degree T in the token-level policy shifts. Consequently, although sequence-level importance weighting provides an unbiased estimator of the expected reward, its variance may outweigh the benefits of unbiasedness, rendering direct application unsuitable under large policy shifts.

5.2 SECOND-ORDER TAYLOR CORRECTION

Correction Formulation. When $k_t \equiv 1$, the surrogate objective $L_{\pi_{\text{old}}}(\pi_\theta)$ in Equation 2 can be interpreted as a first-order Taylor approximation of the expected reward $\mathcal{L}(\pi_\theta)$ in Equation 1:

$$\begin{aligned}\mathcal{L}(\pi_\theta) &\approx \textstyle\sum_{t,\Delta_t} \frac{\partial \mathcal{L}(\pi_\theta)}{\partial \pi_\theta(a_t|a_{0:t},s)}\big|_{\pi_{\text{old}}} \Delta_t + \mathcal{L}(\pi_{\text{old}}) \\ &= \mathbb{E}_{\pi_{\text{old}}}\left[\textstyle\sum_t \omega_t r\right] + (1-T)\mathcal{L}(\pi_{\text{old}}) \overset{k_t \equiv 1}{=} L_{\pi_{\text{old}}}(\pi_\theta),\end{aligned} \tag{11}$$

where $\Delta_t = \pi_\theta(a_t|a_{0:t},s) - \pi_{\text{old}}(a_t|a_{0:t},s)$. This interpretation extends to all algorithms with constant $k_t \equiv k$, since their objectives $L_{\pi_{\text{old}}}(\pi_\theta)$ differ only by the positive scaling factor k and therefore have the same optimizer. Consequently, algorithms such as DAPO and Dr-GRPO can be viewed as maximizing a first-order Taylor approximation of the expected reward.

Moreover, thanks to the non-revisiting structure of MDP in LLM generation ($a_{0:t} \neq a_{0:t'}$ whenever $t \neq t'$), the second-order approximation of the expected reward also has a simple form:

$$\begin{aligned}\mathcal{L}(\pi_\theta) - \mathcal{L}(\pi_{\text{old}}) - \textstyle\sum_{t,\Delta_t} \frac{\partial \mathcal{L}(\pi_\theta)}{\partial \pi_\theta(a_t|a_{0:t},s)}\big|_{\pi_{\text{old}}} \Delta_t &\approx \frac{1}{2}\textstyle\sum_{t,k,\Delta_t,\Delta_k} \frac{\partial^2 \mathcal{L}(\pi_\theta)}{\partial \pi_\theta(a_t|a_{0:t},s)\partial \pi_\theta(a_k|a_{0:k},s)}\big|_{\pi_{\text{old}}} \Delta_t \Delta_k \\ &= \frac{1}{2}\mathbb{E}_{\pi_{\text{old}}}\left[\textstyle\sum_t \sum_{k\neq t} (\omega_t - 1)(\omega_k - 1)\, r\right].\end{aligned} \tag{12}$$

This term requires only a single Monte Carlo sample and can therefore reuse the same rollouts in existing asynchronous methods.

Approximation Error. Equation 11 shows that, when k_t is constant, maximizing the surrogate objective $L_{\pi_{\text{old}}}(\pi_\theta)$ is equivalent to maximizing the first-order Taylor approximation of $\mathcal{L}(\pi_\theta)$ around π_{old}. Accordingly, the off-policy error ϵ_{off} is the absolute Peano remainder and thus is second order in the policy shift. The second-order Taylor term in Equation 12 therefore further corrects the first-order approximation error, reducing the remainder to third order in the policy shift and yielding a more accurate approximation under small policy shifts. The same conclusion follows more explicitly from the Lagrange remainder.

Note that the divergence used in the above error analysis differs from the TV divergence in Equation 4, so the tightness of the two bounds cannot be directly compared. This does not affect our analysis, whose purpose is to show that the second-order Taylor correction further reduces the off-policy error rather than to derive a tighter upper bound.

Variance. Although the second-order approximation still introduces approximation error compared with sequence-level importance sampling, its variance is much smaller. Applying Equation 9 sequentially to the two importance-ratio terms in Equation 12, we obtain the following variance scaling:

$$\operatorname{Var}_{\pi_{\text{old}}}\left[\textstyle\sum_t \sum_{k\neq t} (\omega_t - 1)(\omega_k - 1)\, r\right] = O\left(\mathbb{E}_{\pi_{\text{old},t,k}}\left[(1 + K_t \cdot \mathbb{D}_t)(1 + K_k \cdot \mathbb{D}_k)\right]\right). \tag{13}$$

Thus, the second-order approximation has variance quadratic in the token-level policy shifts, which is lower than that of full sequence-level importance sampling but higher than the linear variance of the first-order approximation. Therefore, its correction should be disabled under large policy shifts.

5.3 BIAS-CORRECTED YIELDED POLICY OPTIMIZATION

Taking into account the unbiased importance-sampling correction, the low-variance first-order approximation, and the second-order approximation that balances bias and variance, we propose Bias-Corrected Yielded Policy Optimization (BYPO), which dynamically selects the appropriate correction term. Since the variance is primarily dominated by samples with large correction magnitudes, rather than necessarily by samples exhibiting large policy shifts at every token, we directly use the absolute magnitude of the correction term as the criterion for correction selection. We further retain PPO-style clipping to control policy shifts and use the advantage function for variance reduction, yielding the following BYPO objective:

$$\mathcal{L}_{\text{BYPO}} = \frac{1}{N}\textstyle\sum_{i,t} \min\left(\omega_{i,t}\tilde{A}_{i,t}, \operatorname{clip}(\omega_{i,t}, \tau_l, \tau_h)\,\tilde{A}_{i,t}\right), \tag{14}$$

$$\tilde{A}_{i,t} = (1 + w_{\text{bypo}} \operatorname{sg}[\delta_{i,t}])\, A_i, \tag{15}$$

where $\omega_{i,t} = \frac{\pi_\theta(a_{i,t}|a_{i,0:t},s)}{\pi_{\text{old}}(a_{i,t}|a_{i,0:t},s)}$ is the importance weight of the t-th token in the i-th response, N is the total number of tokens, and A_i is the advantage of the i-th response. By default, A_i is computed using the GRPO advantage estimator, although it can be replaced by other advantage estimators. Since the total-differential identity $\nabla\phi(\omega_{0:T}) = \nabla \sum_t \omega_t \, \mathrm{sg}[\partial_{\omega_t}\phi(\omega_{0:T})]$ holds for any differentiable function ϕ, the stop-gradient operator $\mathrm{sg}[\cdot]$ in Equation 15 introduces the gradient of the correction term without explicitly including the correction term itself. The two formulations are equivalent from an optimization perspective, while the former makes the correction behavior more transparent and can be more naturally combined with PPO-style clipping. Note that the $\min$ and clip operations correspond to standard PPO-style clipping. They control the policy shift by masking the gradients of outlier samples, thereby allowing BYPO to constrain policy changes while optimizing the corrected objective estimate. The coefficient w_{bypo} controls the strength of the correction, while τ_l and τ_h are the PPO-style clipping thresholds. The BYPO correction term $\delta_{i,t}$ is given by

$$\delta_{i,t} = \begin{cases} \dfrac{\varphi_i^{\text{is}}}{\omega_{i,t}} - 1, & \left|\varphi_i^{\text{is}} - 1\right| \le \tau_{\text{is}}, \\ \varphi_i^{\text{t2}} - \omega_{i,t} + 1, & \left|\varphi_i^{\text{is}} - 1\right| > \tau_{\text{is}},\ \left|\varphi_i^{\text{t2}}\right| \le \tau_{\text{t2}}, \\ 0, & \text{otherwise}, \end{cases} \tag{16}$$

$$\varphi_i^{\text{is}} = \textstyle\prod_k \omega_{i,k}, \qquad \varphi_i^{\text{t2}} = \textstyle\sum_k (\omega_{i,k} - 1). \tag{17}$$

This construction ensures that $\delta_{i,t}A_i$ is exactly the partial derivative of the selected correction term with respect to $\omega_{i,t}$. Clearly, the expected gradient of $\mathcal{L}_{\text{BYPO}}$ interpolates between the gradients of $L_{\pi_{\text{old}}}(\pi_\theta)$ and the fully corrected objective through w_{bypo}, therefore retaining low variance without reshaping the correction. The piecewise definition of $\delta_{i,t}$ further enables fine-grained variance control through dynamic correction selection. Since $\mathbb{E}[\mathbb{I}X + (1-\mathbb{I})Y] = \mathbb{E}[\mathbb{I}]\,\mathbb{E}[X] + \mathbb{E}[1-\mathbb{I}]\,\mathbb{E}[Y] + \mathrm{Cov}(\mathbb{I}, X - Y)$ holds for any selection indicator $\mathbb{I}$ and random variables X and Y, the selection-induced bias is characterized by $\mathrm{Cov}(\mathbb{I}, X - Y)$. In BYPO, the selection indicator depends on the magnitude rather than the sign of the correction, suggesting only weak dependence on the correction term and thus little additional bias. Empirically, the correlation coefficients between each selection indicator and its corresponding correction term consistently remain within $[-0.02, 0.02]$, supporting this hypothesis.

6 Experimental Setup

6.0.1 RLVR Experimental Setup

Training. To cover diverse settings, we train different base models on different datasets. Specifically, we randomly sample 32K questions from DeepMath-103K (He et al., 2025) to train Qwen2.5-7B (Bai et al., 2023), use DAPO-17K (Yu et al., 2026) to train Qwen3-4B (Yang et al., 2025), and use the Competition Math dataset (Hendrycks et al., 2021) to train Llama3.2-1B-Instruct (Dubey et al., 2024). All experiments are conducted using the Verl framework (Sheng et al., 2025). In the Qwen2.5-7B experiment, each rollout stage generates 16 responses for each of 256 prompts, and the resulting data are used for the subsequent 8 optimization iterations with a learning rate of $1\mathrm{e}{-6}$. In the other two experiments, the group size is set to 8, and rollouts generated from 256 prompts are used for 4 parameter updates. Across all three experiments, BYPO adopts the same clipping thresholds as DAPO (Yu et al., 2026). We set $w_{\text{bypo}} = 0.1$, $\tau_{\text{is}} = 0.05$, and $\tau_{\text{t2}} = 1.0$ in all experiments.

Baselines. For a comprehensive comparison, we include GRPO (Shao et al., 2024), Dr-GRPO (Liu et al., 2025), DAPO (Yu et al., 2026), RLOO (Ahmadian et al., 2024), and REINFORCE++ (Hu et al., 2025) as baselines without correction terms, along with VESPO (Shen et al., 2026) and TOPR (Roux et al., 2025), which apply nonlinear transformations to sequence-level importance weights. Because dynamic sampling substantially increases training time while providing only limited performance gains, we disable it in the main experiments while retaining all other DAPO components. We evaluate DAPO with dynamic sampling separately.

Evaluation. For comprehensive evaluation, we use 10 widely adopted mathematical reasoning benchmarks, including AIME24, AIME25, GSM8K (Cobbe et al., 2021), MATH500 (Hendrycks et al., 2021), AMC23, Olympiad Bench (He et al., 2024), MinervaMath (Lewkowycz et al., 2022),

Table 1: Main RLVR results. Qwen2.5-7B, Qwen3-4B, and Llama3.2-1B-Instruct are trained for one epoch on a subset of Deepmath-103K, DAPO-17K, and Competition Math, respectively. We report accuracy on standard mathematical reasoning benchmarks, as well as the average accuracy across all benchmarks (AVG). "Base" refers to the metrics of the base model. **Bold** numbers indicate the best results for each metric.

Method	AIME24 ↑	AIME25 ↑	GSM8k ↑	MATH500 ↑	AMC23 ↑	Olympiad ↑	Minerva ↑	HMMT ↑	AVG ↑
Qwen2.5-7B									
Base	4.00	3.33	59.06	41.40	26.00	18.22	9.19	3.11	20.54
RLOO	11.87	11.47	91.13	71.80	53.67	33.28	23.16	2.83	37.40
RF++	12.53	9.20	90.30	73.60	53.67	34.47	24.63	4.39	37.85
GRPO	15.20	10.13	89.46	72.80	56.67	35.66	**26.47**	4.61	38.88
Dr-GRPO	15.47	9.87	91.28	74.20	55.67	35.51	23.90	4.94	38.85
DAPO	14.93	8.67	91.36	74.00	**59.17**	36.26	23.90	4.00	39.03
VESPO	14.00	9.87	91.05	72.80	58.17	**38.19**	23.16	**5.00**	39.03
TOPR	10.13	9.33	91.74	**75.40**	**59.17**	35.96	24.26	4.28	38.78
BYPO	**16.27**	**13.87**	**91.96**	74.20	57.67	**38.19**	25.00	4.11	**40.16**
Qwen3-4B									
Base	38.93	26.53	93.48	83.40	74.17	47.40	29.60	18.55	51.51
RLOO	53.87	45.33	93.63	88.20	87.00	57.36	32.17	26.55	60.51
RF++	54.67	41.73	93.93	87.20	84.17	53.34	31.99	24.71	58.97
GRPO	55.73	44.13	92.95	**89.00**	88.67	56.46	31.99	27.44	60.80
Dr-GRPO	54.40	**46.13**	93.18	88.40	87.33	55.57	**33.64**	27.10	60.72
DAPO	55.60	44.53	93.93	87.80	89.00	57.21	31.62	27.32	60.88
VESPO	54.93	40.40	**94.31**	87.60	88.00	56.46	29.41	**27.88**	59.88
TOPR	56.00	44.40	93.25	88.60	87.50	57.36	31.43	27.65	60.77
BYPO	**57.46**	45.47	93.56	88.80	**91.17**	**57.65**	31.62	27.77	**61.69**
Llama3.2-1B-Instruct									
Base	0.53	0.00	4.85	14.20	5.33	4.16	1.47	0.22	3.84
RLOO	0.80	**0.53**	53.98	35.20	18.50	8.02	**7.17**	0.78	15.62
RF++	1.73	0.27	52.24	30.00	15.67	6.69	5.33	1.22	14.14
GRPO	1.47	0.40	53.45	31.60	17.33	6.69	**7.17**	0.28	14.80
Dr-GRPO	1.73	0.00	53.53	32.00	19.17	7.13	5.33	0.83	14.97
DAPO	2.53	0.00	53.75	34.00	15.50	7.43	5.15	0.61	14.87
VESPO	2.67	0.13	**56.25**	33.00	19.67	7.28	4.23	**1.06**	15.54
TOPR	2.00	0.27	55.72	31.60	18.67	8.02	6.43	0.72	15.43
BYPO	**3.20**	0.27	54.74	**35.60**	**23.00**	**8.17**	6.07	0.33	**16.42**

HMMT 2402, HMMT 2502, and HMMT 2511 (Dekoninck et al., 2026). The average score over the three HMMT subsets is reported as the final HMMT result.

6.0.2 RLHF Experimental Setup

Training. Following Han et al. (2026), we use TRL (von Werra et al., 2020) to train Llama-3.1-8B-Instruct on the UltraFeedback dataset (Cui et al., 2023), with GRM-Llama3.2-3B-rewardmodel-ft (Yang et al., 2024) as the reward model. We set the generation batch size to 96, the mini-batch size to 32, and the learning rate to $1e{-5}$. BYPO uses the same hyperparameters as in the RLVR experiments.

Baselines. We compare with classical RLHF methods, including PPO (Achiam et al., 2023), DPO (Rafailov et al., 2023), IPO (Azar et al., 2024), GRPO (Shao et al., 2024), and RLOO (Ahmadian et al., 2024).

Evaluation. Following Han et al. (2026), we evaluate on AlpacaEval (Dubois et al., 2024) using the version-1 win rate (WR_1), the version-2 win rate (WR_2), and the length-controlled win rate (LC_2). On Arena-Hard (Li et al., 2024), we report the win rates against Gemini-2.0-Flash and GPT-4-0613, respectively.

7 Experimental Results

7.1 Main Comparison with Baselines

RLVR Results. The RLVR results are reported in Table 1. BYPO achieves the highest average score across all three settings and outperforms DAPO, which differs from BYPO only in omitting the correction term, on most benchmarks. Notably, the gap between the strongest and weakest baselines remains below two points in every setting, whereas BYPO further improves upon the strongest baseline by approximately one point, demonstrating a substantial gain. Among the baselines without correction terms, DAPO and Dr-GRPO generally achieve higher average scores, possibly because they better preserve the first-order approximation without introducing additional bias. VESPO and TOPR outperform DAPO on the 1B model, whose generated responses average approximately 0.5K tokens, but fall behind DAPO on the 7B and 4B models, which produce longer responses. We attribute this trend to the more severe off-policy mismatch associated with longer responses, which may amplify the distortion introduced by nonlinear transformations of the importance weights.

RLHF Results. The RLHF results are shown in Table 2. BYPO outperforms existing methods on all metrics, with more than an 8-point improvement over the best baseline on the length-controlled win rate (LC_2) and Arena-Hard, where model differences are more distinguishable (Li et al., 2024). The advantage in length-controlled win rate indicates that BYPO does not obtain higher rewards merely by generating longer responses. We also observe that DPO and IPO rank higher on AlpacaEval than on Arena-Hard, which we attribute to the tendency of preference-optimization methods to prioritize scenarios more closely aligned with AlpacaEval.

Table 2: Experimental results on RLHF. "Base" denotes the base model, and **bold** indicates the best result. On AlpacaEval, WR_1 and WR_2 are win rates on versions 1.0 and 2.0 judged by Llama3-70B-Instruct, while LC_2 is the length-controlled win rate on version 2.0. On Arena Hard, GPT-4 and Gemini-2 denote win rates against GPT-4-0613 (v0.1) and Gemini-2.0-Flash (v2.0), respectively.

Method	Alpaca Eval			Arena Hard	
	$\mathrm{WR}_1 \uparrow$	$\mathrm{WR}_2 \uparrow$	$\mathrm{LC}_2 \uparrow$	GPT-4 $\uparrow$	Gemini-2 $\uparrow$
Base	92.29	42.13	46.72	36.20	17.92
PPO	94.52	45.32	52.50	49.80	20.19
DPO	95.30	48.54	55.78	42.20	19.65
IPO	94.89	48.57	54.53	41.20	19.79
GRPO	96.74	50.56	57.20	63.40	30.34
RLOO	96.90	53.26	57.29	60.60	31.82
BYPO	**97.56**	**54.89**	**65.70**	**77.60**	**44.12**

7.2 How Much Degradation Does BYPO Mitigate?

To quantify how much off-policy degradation BYPO mitigates, we use fully synchronous DAPO as a reference and plot training-set accuracy in Figure 2. Synchronous DAPO improves much faster than asynchronous DAPO. Early in training, BYPO-corrected DAPO underperforms the uncorrected version, likely due to the higher variance introduced by correction. After about 70 steps, however, BYPO gradually surpasses uncorrected DAPO and approaches synchronous DAPO. Stronger correction similarly yields lower early-stage but higher late-stage accuracy, supporting a higher-variance, lower-error trade-off. Thus, $w_{\mathrm{bypo}} = 0.1$ may not be universally optimal, and stronger correction may become more beneficial with longer training. This also motivates a potentially valuable direction for future work: dynamically scheduling w_{bypo}.

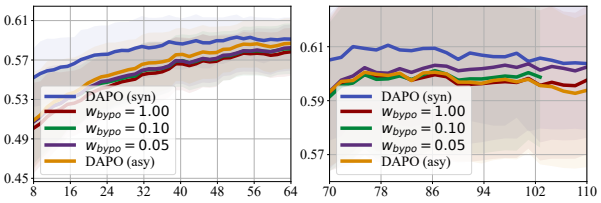


Figure 2: Training accuracy curves for Qwen2.5-7B. The x-axis shows training steps, and the y-axis shows training-set accuracy. "DAPO (syn)" and "DAPO (asy)" denote synchronous and asynchronous DAPO, respectively. The three w_{bypo} curves show BYPO results with the corresponding correction coefficient.

7.3 Generalization across Algorithms

BYPO makes no assumptions about the advantage estimator or the data distribution and, therefore, should be applicable to most policy optimization methods. To verify this, based on the Qwen2.5-7B experiment, we replace the objectives of dynamic-sampling DAPO, RLOO, REINFORCE++, and Dr-GRPO with the BYPO objective while keeping their advantage estimators unchanged. The final results are shown in Table 3, where BYPO improves the performance of all algorithms. In particular, BYPO-corrected dynamic-sampling DAPO achieves the highest score, outperforming BYPO without dynamic sampling in Table 1, indicating that BYPO does not sacrifice the performance gains brought by better data selection. We also observe the largest improvement on RLOO, possibly because its cleaner advantage estimates allow BYPO to better realize the benefits of unbiasedness.

Table 3: Average accuracy across all benchmarks before and after applying BYPO to different algorithms on Qwen2.5-7B. The first and second rows report results before and after BYPO correction, respectively. DAPO† denotes DAPO with dynamic sampling.

BYPO	DAPO†	RLOO	RF++	Dr-GRPO
✗	39.25	37.40	37.85	38.85
✓	40.72 +1.5	39.80 +2.4	39.40 +1.6	39.04 +0.2

7.4 Ablation Study

To study the effects of different correction terms and hyperparameters, we conduct ablations on Qwen2.5-7B, with results in Table 4. Using only importance sampling (Setting 2) or the second-order Taylor correction (Setting 4) degrades performance due to high variance, with importance weights exhibiting greater variance. The bounded second-order Taylor correction alone (Setting 5) achieves the second-best score but slightly underperforms Setting 1 because it does not fully exploit the unbiasedness of importance weighting. All BYPO variants (Settings 1, 3, and 5–9) outperform uncorrected DAPO, demonstrating robustness. Their gains vary with correction weights and thresholds, reflecting different bias–variance trade-offs. Although BYPO is somewhat sensitive to hyperparameters, Setting 1 performs consistently well across experiments, suggesting that a single carefully chosen configuration can generalize across settings and may point to a principled way of selecting correction thresholds.

Table 4: Ablation results for Qwen2.5-7B. ρ_{is} and ρ_{t2} are the proportions of responses corrected via importance sampling and second-order Taylor expansion, respectively. σ^2 is the variance of the correction terms $\delta_{i,t}$ before scaling by w_{bypo}. Underlined values outperform DAPO, and **bold** values indicate the best results.

Setting	1	2	3	4	5	6	7	8	9
w_{bypo}	0.10	0.10	0.10	0.10	0.10	0.05	0.15	0.20	1.00
τ_{is}	0.05	$+\infty$	0.10	0.00	0.00	0.05	0.05	0.05	0.05
τ_{t2}	1.00	0.00	1.00	$+\infty$	1.00	1.00	0.70	0.70	1.00
σ^2	0.20	0.70	0.20	0.52	0.17	0.20	0.11	0.11	0.21
ρ_{is}	0.14	1.00	0.21	0.00	0.00	0.14	0.14	0.15	0.15
ρ_{t2}	0.74	0.00	0.68	1.00	0.87	0.74	0.61	0.61	0.74
Acc	**40.2**	38.4	39.0	38.5	40.1	39.5	39.9	39.2	39.9

8 Conclusion

In this paper, we propose BYPO, which adaptively uses sequence-level importance weights or second-order Taylor correction weights to correct the objective in asynchronous token-level RL for LLMs. BYPO introduces almost no additional computational overhead, and improves the performance of trained models by reducing off-policy error. BYPO achieves the strongest performance gains in both RLVR and RLHF settings, and is shown to correct 83.70% of the performance degradation caused by off-policy error in the RLVR setting. Although BYPO improves final training performance, it slightly slows down early-stage training, which we attribute to the variance introduced by correction. A promising direction for future work is to develop mechanisms that adaptively adjust the correction coefficient, so as to better balance convergence speed and final performance.

AI USE STATEMENT

In this work, we used generative AI tools solely for translation and grammatical editing. They were not used for any core research activities, including theoretical modeling, mathematical formulation, experimental design, method implementation, data processing, or manuscript drafting. Tasks involving synthetic dataset generation or qualitative data analysis are not applicable to this work. We carefully reviewed all AI-assisted content for correctness and take full responsibility for the final content of this work, including any text, claims, or artifacts produced with the assistance of generative AI.

REPRODUCIBILITY STATEMENT

We provide sufficient details to facilitate reproduction of our theoretical and empirical results. All major theoretical results are presented in the main text following the derivation flow, with complete proofs provided in Appendix A. The main experimental results and training configurations are reported in the experimental section, while additional implementation details are provided in Appendix B. We use the open-source Verl and TRL reinforcement learning frameworks, and all models and datasets used in our experiments are publicly available. Our code will be released upon publication.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

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.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pp. 4447–4455. PMLR, 2024.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Shreyas Chaudhari, Pranjal Aggarwal, Vishvak Murahari, Tanmay Rajpurohit, Ashwin Kalyan, Karthik Narasimhan, Ameet Deshpande, and Bruno Castro da Silva. RLHF deciphered: A critical analysis of reinforcement learning from human feedback for LLMs. *ACM Computing Surveys*, 58 (2):1–37, 2025.

Xiangxiang Chu, Hailang Huang, Xiao Zhang, Fei Wei, and Yong Wang. GPG: A simple and strong reinforcement learning baseline for model reasoning. *arXiv preprint arXiv:2504.02546*, 2025.

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

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. UltraFeedback: Boosting language models with high-quality feedback, 2023.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, et al. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025.

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

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. Length-controlled AlpacaEval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Wei Han, Yuanxing Liu, Mingda Li, Ruiyu Xiao, Weinan Zhang, and Ting Liu. MRPO: Magnitude-regularized policy optimization via l1 constraints. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=e2xQL4BWrY.

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.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, et al. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. *arXiv preprint arXiv:2504.11456*, 2025.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In J. Vanschoren and S. Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/be83ab3ecd0db773eb2dc1b0a17836a1-Paper-round2.pdf.

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

Tianhao Hu, Xiangcheng Liu, Youshao Xiao, Yang Zheng, Xuan Huang, Jinrui Ding, Yufei Zhang, Tao Liang, Hongyu Zang, Quan Chen, et al. DORA: A scalable asynchronous reinforcement learning system for language model training. *arXiv preprint arXiv:2604.26256*, 2026.

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

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E Gonzalez, and Ion Stoica. From crowdsourced data to high-quality benchmarks: Arena-Hard and benchbuilder pipeline. *arXiv preprint arXiv:2406.11939*, 2024.

Jiaqing Liang, Jinyi Han, Xinyi Wang, Zishang Jiang, Chengyuan Xiong, Boyu Zhu, Jie Shi, Weijia Li, Tingyun Li, and Yanghua Xiao. KW-R1: A simple implementation of the GRPO algorithm. https://github.com/lsdefine/simple_GRPO, 2025.

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.

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.

Nicolas Le Roux, Marc G Bellemare, Jonathan Lebensold, Arnaud Bergeron, Joshua Greaves, Alex Fréchette, Carolyne Pelletier, Eric Thibodeau-Laufer, Sándor Toth, and Sam Work. Tapered off-policy reinforce: Stable and efficient reinforcement learning for LLMs. *arXiv preprint arXiv:2503.14286*, 2025.

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.

Guobin Shen, Chenxiao Zhao, Xiang Cheng, Lei Huang, and Xing Yu. Vespo: Variational sequence-level soft policy optimization for stable off-policy llm training. *arXiv preprint arXiv:2602.10693*, 2026.

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. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.

Fahim Tajwar, Guanning Zeng, Yueer Zhou, Yuda Song, Daman Arora, Yiding Jiang, Jeff Schneider, Ruslan Salakhutdinov, Haiwen Feng, and Andrea Zanette. Maximum likelihood reinforcement learning. *arXiv preprint arXiv:2602.02710*, 2026.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, et al. KIMI K2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025.

Minghao Tian, Yunfei Xie, and Chen Wei. How off-policy can GRPO be? Mu-GRPO for efficient LLM reinforcement learning. *arXiv preprint arXiv:2605.17570*, 2026.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. TRL: Transformer reinforcement learning. `https://github.com/huggingface/trl`, 2020.

Chaoyi Wang and Yuan Sun. MSEED: Human preference dataset for multidimensional safety enhancement and evaluation of large language models. *Data Intelligence*, 7(4):1243–1269, 2025. doi: 10.3724/2096-7004.di.2025.0080.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Rui Yang, Ruomeng Ding, Yong Lin, Huan Zhang, and Tong Zhang. Regularizing hidden states enables learning generalizable reward model for LLMs. *Advances in Neural Information Processing Systems*, 37:62279–62309, 2024.

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. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

Chujie Zheng, Kai Dang, Bowen Yu, Mingze Li, Huiqiang Jiang, Junrong Lin, Yuqiong Liu, An Yang, Jingren Zhou, and Junyang Lin. Stabilizing reinforcement learning with llms: Formulation and practices. *arXiv preprint arXiv:2512.01374*, 2025a.

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*, 2025b.

Haizhong Zheng, Jiawei Zhao, and Beidi Chen. Prosperity before collapse: How far can off-policy rl reach with stale data on llms? *arXiv preprint arXiv:2510.01161*, 2025c.

A PROOFS

Theorem 1 (Policy-shift constraints are insufficient)**.** *When the reward and k_t are bounded, Equation 5 holds.*

Proof. Assume that action a_t^* has the optimal action value, i.e., $Q_t^{\pi_{\text{old}}}(a_t^*, a_{0:t}, s) = \max_a Q_t^{\pi_{\text{old}}}(a, a_{0:t}, s) = Q_t^*$, and that action a_t^- has the minimum action value, i.e., $Q_t^{\pi_{\text{old}}}(a_t^-, a_{0:t}, s) = \min_a Q_t^{\pi_{\text{old}}}(a, a_{0:t}, s) = Q_t^-$. Then the following derivation holds:

$$\begin{aligned}
L_{\pi_{\text{old}}}(\pi_\theta) - L_{\pi_{\text{old}}}(\pi_{\text{old}}) &= \mathbb{E}_{\pi_{\text{old}}}\left[\textstyle\sum_t k_t \left(\mathbb{E}_{\pi_\theta}\left[Q_t^{\pi_{\text{old}}}|a_{0:t}, s\right] - \mathbb{E}_{\pi_{\text{old}}}\left[Q_t^{\pi_{\text{old}}}|a_{0:t}, s\right]\right)\right] \\
&= \mathbb{E}_{\pi_{\text{old}}}\left[\textstyle\sum_t k_t \left(\sum_{a_t} \Delta_t Q_t^{\pi_{\text{old}}}\right)\right] \\
&= \mathbb{E}_{\pi_{\text{old}}}\left[\textstyle\sum_t k_t \sum_{\Delta_t>0} \Delta_t Q_t^{\pi_{\text{old}}}\right] + \mathbb{E}_{\pi_{\text{old}}}\left[\textstyle\sum_t k_t \sum_{\Delta_t<0} \Delta_t Q_t^{\pi_{\text{old}}}\right] \\
&\le \mathbb{E}_{\pi_{\text{old}}}\left[\textstyle\sum_t k_t Q_t^* \sum_{\Delta_t>0} \Delta_t\right] + \mathbb{E}_{\pi_{\text{old}}}\left[\textstyle\sum_t k_t Q_t^- \sum_{\Delta_t<0} \Delta_t\right] \\
&= \left[\textstyle\sum_t k_t (Q_t^* - Q_t^-) \mathbb{D}_{\text{TV}}(\pi_\theta(\cdot|a_{0:t}, s) \| \pi_{\text{old}}(\cdot|a_{0:t}, s))\right] \\
&\le Q^\Delta \mathbb{E}_{\pi_{\text{old}}}\left[\textstyle\sum_t \mathbb{D}_{\text{TV}}\left(\pi_\theta(\cdot|a_{0:t}, s) \,\|\, \pi_{\text{old}}(\cdot|a_{0:t}, s)\right)\right],
\end{aligned} \tag{18}$$

where $\Delta_t = \pi_\theta(a_t|a_{0:t}, s) - \pi_{\text{old}}(a_t|a_{0:t}, s)$, and:

$$\mathbb{D}_{\text{TV}}(\pi_\theta(\cdot|a_{0:t}, s) \| \pi_{\text{old}}(\cdot|a_{0:t}, s)) = \frac{1}{2}\textstyle\sum_{a_t} |\Delta_t| = \sum_{\Delta_t>0} \Delta_t = -\sum_{\Delta_t<0} \Delta_t. \tag{19}$$

□

Lemma 1 (Token-level importance-weight variance identity)**.** *Equation 8 holds.*

Proof. By the definition of covariance, we have:

$$\begin{aligned}
\mathbb{E}_{\pi_{\text{old}}(a_t|s,a_{0:t})}[\omega_t^2 f^2] &= \mathbb{E}_{\pi_\theta(a_t|s,a_{0:t})}[\omega_t f^2] \\
&= \mathbb{E}_{\pi_\theta(a_t|s,a_{0:t})}[\omega_t]\mathbb{E}_{\pi_\theta(a_t|s,a_{0:t})}[f^2] + \text{Cov}_{\pi_\theta(a_t|s,a_{0:t})}\left(\omega_t, f^2\right) \\
&= (1 + \mathbb{D}_t)\mathbb{E}_{\pi_\theta(a_t|s,a_{0:t})}[f^2] + \text{Cov}_{\pi_\theta(a_t|s,a_{0:t})}\left(\omega_t, f^2\right) \\
&= C_t(1 + \mathbb{D}_t)\mathbb{E}_{\pi_\theta(a_t|s,a_{0:t})}[f^2].
\end{aligned} \tag{20}$$

Under the setting considered in the main text, C_t can be shown to be strictly positive:

$$C_t = 1 + \frac{\text{Cov}_{\pi_\theta(a_t|s,a_{0:t})}\left(\omega_t, f^2\right)}{\mathbb{E}_{\pi_\theta(a_t|s,a_{0:t})}[\omega_t]\,\mathbb{E}_{\pi_\theta(a_t|s,a_{0:t})}[f^2]} = \frac{\mathbb{E}_{\pi_\theta(a_t|s,a_{0:t})}[\omega_t f^2]}{\mathbb{E}_{\pi_\theta(a_t|s,a_{0:t})}[\omega_t]\,\mathbb{E}_{\pi_\theta(a_t|s,a_{0:t})}[f^2]}. \tag{21}$$

Since both ω_t and f^2 are nonnegative, have nonzero expectations, and are not simultaneously zero ($\mathbb{E}_{\pi_\theta}[f^2] \neq 0$), C_t is strictly positive and bounded. □

Lemma 2 (Importance-weight variance in sequence-level settings)**.** *Equation 9 holds.*

Proof. By the definition of variance, we directly upper-bound it by the second moment:

$$\text{Var}_{\pi_{\text{old}}}\left[(\omega_t - 1)f\right] \le \mathbb{E}_{\pi_{\text{old}}}\left[(\omega_t - 1)^2 f^2\right]. \tag{22}$$

Define the auxiliary variable $g_t = \mathbb{E}_{\pi_{\text{old}}(a_{t+1:T}|a_{0:t+1},s)}[f^2]$. By applying Lemma 1, we obtain:

$$\begin{aligned}
\mathbb{E}_{\pi_{\text{old}}}\left[(\omega_t - 1)^2 f^2\right] &= \mathbb{E}_{\pi_{\text{old}}(a_{0:t}|s)}\left[\mathbb{E}_{\pi_{\text{old}}(a_t|a_{0:t},s)}\left[(\omega_t - 1)^2 g_t\right]\right] \\
&= \mathbb{E}_{\pi_{\text{old}}(a_{0:t}|s)}\left[\mathbb{E}_{\pi_{\text{old}}}\left[\omega_t^2 g_t\right] - 2\mathbb{E}_{\pi_{\text{old}}}\left[\omega_t g_t\right] + \mathbb{E}_{\pi_{\text{old}}}\left[g_t\right]\right] \\
&= \mathbb{E}_{\pi_{\text{old}}(a_{0:t}|s)}\left[C_t(1 + \mathbb{D}_t)\mathbb{E}_{\pi_\theta}[g_t] - 2\mathbb{E}_{\pi_\theta}\left[g_t\right] + \mathbb{E}_{\pi_{\text{old}}}\left[g_t\right]\right] \\
&= \mathbb{E}_{\pi_{\text{old}}(a_{0:t}|s)}\left[\left(C_t(1 + \mathbb{D}_t) - 2\right)\mathbb{E}_{\pi_\theta}[g_t] + \mathbb{E}_{\pi_{\text{old}}}[g_t]\right] \\
&= \mathbb{E}_{\pi_{\text{old}}(a_{0:t}|s)}\left[\left(\left(C_t(1 + \mathbb{D}_t) - 2\right)R_t + 1\right)\mathbb{E}_{\pi_{\text{old}}}[f^2]\right],
\end{aligned} \tag{23}$$

where $R_t = \frac{\mathbb{E}_{\pi_\theta}[g_t]}{\mathbb{E}_{\pi_{\text{old}}}[g_t]}$. By the Cauchy–Schwarz inequality, we can show that each variable in the above expression is bounded:

$$|R_t - 1| = \left| \frac{\mathbb{E}_{\pi_\theta}[g_t]}{\mathbb{E}_{\pi_{\text{old}}}[g_t]} - 1 \right| = \frac{|\mathbb{E}_{\pi_{\text{old}}}[(\omega_t - 1)g_t]|}{\mathbb{E}_{\pi_{\text{old}}}[g_t]} \leq \frac{\sqrt{\mathbb{D}_t \mathbb{E}_{\pi_{\text{old}}}[g_t^2]}}{\mathbb{E}_{\pi_{\text{old}}}[g_t]}. \tag{24}$$

Rearranging the terms completes the proof. □

Lemma 3 (First-order Taylor approximation of the RL objective)**.** *Equation 11 holds.*

Proof. By definition, we only need to compute the partial derivative of $\mathcal{L}(\pi)$ with respect to $\pi(a_t|a_{0:t}, s)$:

$$\frac{\partial \mathcal{L}(\pi)}{\partial \pi(a_t|a_{0:t}, s)}\Big|_{\pi_{\text{old}}} = \frac{\partial}{\partial \pi(a_t|a_{0:t}, s)} \sum_{a'_{0:T}} r \prod_t \pi(a'_t|a'_{0:t}, s)\Big|_{\pi_{\text{old}}} = \pi_{\text{old}}(a_{0:t}|s)\mathbb{E}_{\pi_{\text{old}}}[r \mid a_{0:t+1}, s]. \tag{25}$$

We next derive the first-order Taylor approximation:

$$\begin{aligned} \sum_{t, a_{0:t+1}} \Delta_t \frac{\partial \mathcal{L}(\pi)}{\partial \pi(a_t|a_{0:t}, s)}\Big|_{\pi_{\text{old}}} &= \sum_{t, a_{0:t+1}} \pi_{\text{old}}(a_{0:t}|s)\pi_\theta(a_t|a_{0:t}, s)\mathbb{E}_{\pi_{\text{old}}}[r \mid a_{0:t+1}, s] - C(\pi_{\text{old}}) \\ &= \sum_{t, a_{0:t+1}} \omega_t \pi_{\text{old}}(a_{0:t+1}|s)\mathbb{E}_{\pi_{\text{old}}}[r \mid a_{0:t+1}, s] - C(\pi_{\text{old}}) \\ &= \mathbb{E}_{\pi_{\text{old}}}\left[\sum_t \omega_t r\right] - C(\pi_{\text{old}}). \end{aligned} \tag{26}$$

where $C(\pi_{\text{old}})$ is a constant. The proof is completed by removing this constant through a simple translation $\zeta(\pi_{\text{old}})$ defined in Equation 2. □

Theorem 2 (Second-order Taylor approximation of the RL objective)**.** *Equation 12 holds.*

Proof. Similarly, we only need to compute the second-order partial derivatives and the corresponding second-order Taylor expansion. Without loss of generality, assume that $k \leq t$. Then:

$$\begin{aligned} &\frac{\partial^2 \mathcal{L}(\pi)}{\partial \pi(a_t|a_{0:t}, s)\partial \pi(b_k|b_{0:k}, s)}\Big|_{\pi_{\text{old}}} \\ =&\mathbb{I}_{b_{0:k+1}=a_{0:k+1}} \mathbb{I}_{t \neq k} \pi_{\text{old}}(a_{0:k}|s)\pi_{\text{old}}(a_{k+1:t}|a_{0:k+1}, s)\mathbb{E}_{\pi_{\text{old}}}[r \mid a_{0:t+1}, s]. \end{aligned} \tag{27}$$

We then construct the second-order Taylor approximation:

$$\begin{aligned} &\frac{1}{2}\sum_{t,k,a_{0:t+1},b_{0:k+1}} \Delta_t \Delta_k \frac{\partial^2 \mathcal{L}(\pi)}{\partial \pi(a_t|a_{0:t}, s)\partial \pi(b_k|b_{0:k}, s)}\Big|_{\pi_{\text{old}}} \\ =&\frac{1}{2}\sum_{t \neq k, a_{0:t+1}} \Delta_t \Delta_k \pi_{\text{old}}(a_{0:k}|s)\pi_{\text{old}}(a_{k+1:t}|a_{0:k+1}, s)\mathbb{E}_{\pi_{\text{old}}}[r \mid a_{0:t+1}, s] \\ =&\sum_{t > k, a_{0:t+1}} \Delta_t \Delta_k \pi_{\text{old}}(a_{0:k}|s)\pi_{\text{old}}(a_{k+1:t}|a_{0:k+1}, s)\mathbb{E}_{\pi_{\text{old}}}[r \mid a_{0:t+1}, s] \\ =&\sum_{t > k, a_{0:t+1}} (\omega_t - 1)(\omega_k - 1)\pi_{\text{old}}(a_{0:t+1}|s)\mathbb{E}_{\pi_{\text{old}}}[r \mid a_{0:t+1}, s] \\ =&\mathbb{E}_{\pi_{\text{old}}}\left[\sum_{t > k} (\omega_t - 1)(\omega_k - 1) r\right] \\ =&\frac{1}{2}\mathbb{E}_{\pi_{\text{old}}}\left[\sum_{t \neq k} (\omega_t - 1)(\omega_k - 1) r\right] \end{aligned} \tag{28}$$

This completes the proof. □

B Experimental Details

RLVR Experimental Settings. The main training parameters have already been reported in the main text. Here, we provide more detailed training parameters in Table 5.

RLHF Experimental Settings. The main training parameters have already been reported in the main text. Here, we provide more detailed training parameters in Table 6. Note that although two GPUs are used, one is dedicated to hosting the vLLM service and the other is used for training; therefore, only a single GPU is actually used for parameter updates.

Table 5: Common parameters used during training. Algorithm-specific parameters are either provided in the main text or set to their default values.

Parameter	Qwen7B	Qwen4B	Llama1B
kl coef	0	0	0
train batch size	256	256	256
group size	16	8	8
max prompt length	1k	1k	1k
max response length	7k	9k	5k
learning rate	1e-6	1e-6	1e-6
clip ratio low	0.20	0.20	0.20
clip ratio high	0.28	0.28	0.28
ppo mini batch size	32	64	64
lr warmup steps	5	5	5
weight decay	0.1	0.1	0.1
use kl loss	false	false	false
use dynamic bsz	true	true	true
grad clip	1.0	1.0	1.0
ulysses sequence parallel size	2	1	1
tensor model parallel size	2	1	1
gpu memory utilization	0.8	0.8	0.8
number of gpus	4	4	2
training time (hours)	~ 20	~ 65	~ 2

Table 6: Common parameters used during training. Algorithm-specific parameters are either provided in the main text or set to their default values.

Parameter	Value
reward model	GRM
num train epochs	1
gradient accumulation steps	32
num generations	3
per device train batch size	3
lora r	8
lora alpha	16
max completion len	512
beta	0.01
steps per generation	96
learning rate	1e-5
bf16	True
bf16 full eval	True
use vllm	True
number of gpus	2

C Toy Example

To more directly examine the behavior of BYPO, we consider a simple three-step setting in which the policy chooses between two actions at each step. Under π_{old}, the probabilities of the trajectories 000, 001, 010, 011, 100, 101, 110, and 111 are set to 0.18, 0.12, 0.12, 0.08, 0.18, 0.12, 0.12, and 0.08, respectively. Under the current policy, the corresponding probabilities are 0.108, 0.072, 0.072, 0.048, 0.21, 0.315, 0.105, and 0.07. BYPO uses the same hyperparameters as in the main text. The results are shown in Figure 3. BYPO achieves both estimator variance and off-policy estimation error between those of full correction with sequence-level importance weights and no correction. This suggests that, by adjusting the BYPO weight, we can achieve a better error–variance trade-off, reducing off-policy error as much as possible while keeping the variance within a tolerable range, thereby improving training performance.

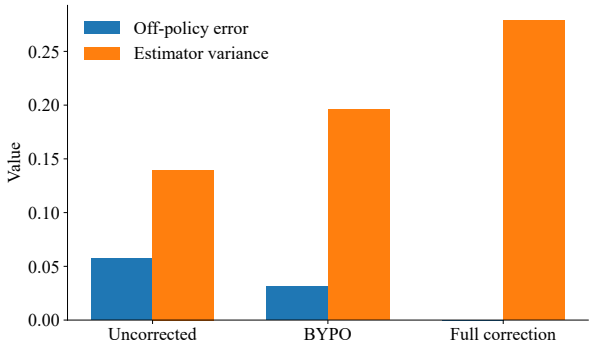


Figure 3: Toy example of BYPO. We consider a simple three-step setting and compute the off-policy estimation error and estimator variance of different offline objective estimators. BYPO achieves a favorable trade-off between estimation error and variance.

Table 7: Ablation study. Results with different BYPO hyperparameter settings under the Qwen2.5-7B experimental setup.

w_{bypo}	τ_{is}	τ_{t2}	AIME24 ↑	AIME25 ↑	GSM8k ↑	MATH500 ↑	AMC23 ↑	Olympiad ↑	Minerva ↑	HMMT ↑
0.10	0.05	1.00	16.27	**13.87**	91.96	74.20	57.67	38.19	25.00	4.11
0.10	$+\infty$	0.00	14.53	8.53	90.75	73.80	53.67	37.00	23.90	**5.28**
0.10	0.10	1.00	15.87	8.00	91.13	73.40	55.67	38.34	24.63	5.22
0.10	0.00	$+\infty$	14.93	12.00	89.84	73.00	52.17	35.66	26.10	4.00
0.10	0.00	1.00	14.67	11.87	91.13	**76.40**	60.50	**39.08**	22.43	4.33
0.05	0.05	1.00	14.80	13.33	91.43	73.60	57.67	37.30	22.79	4.72
0.15	0.05	0.70	14.53	10.00	**92.27**	74.00	**61.33**	38.19	24.26	4.28
0.20	0.05	0.70	14.53	5.73	91.58	73.60	61.00	37.44	**26.47**	3.44
1.00	0.05	1.00	**16.93**	10.93	92.04	75.60	59.00	36.11	23.90	4.33

D More Experimental Results

D.1 Ablation Study

In the main text, we report the overall scores from the ablation study under different hyperparameter settings. Here, we provide the detailed scores for each benchmark, as shown in Table 7.

D.2 BYPO across Different Algorithms

In the main text, we report the overall scores of different algorithms after applying BYPO correction. Here, we provide the complete benchmark-level results. As shown in Table 8, all BYPO-corrected algorithms achieve higher overall scores than their uncorrected counterparts and outperform them on most benchmarks. Notably, all algorithms improve after correction on more challenging benchmarks such as AIME25 and Olympiad, whereas some corrected variants underperform on the relatively easier GSM8K and AMC23 benchmarks. This suggests that the off-policy error primarily corrected by BYPO may have a greater impact on more difficult problems.

D.3 Behavior of Dynamic Correction-Term Selection

To examine the behavior of the correction terms in greater detail, we measure their activation rates in the Qwen2.5-7B experiments, together with their correlation coefficients with the dynamic correction-selection operators. As shown in the main text, these correlations characterize the additional bias introduced by the selection operators. The results are shown in Figure 4. The correction terms remain highly active throughout training, with most activations coming from the second-order

Table 8: Performance comparison of different algorithms before and after applying BYPO correction in the Qwen2.5-7B experiments. We report accuracy on standard mathematical reasoning benchmarks, as well as the average accuracy across all benchmarks (AVG). **Bold** numbers indicate the best result for each metric. DAPO† denotes DAPO with dynamic sampling. After BYPO correction, the algorithms achieve higher overall scores and outperform their uncorrected counterparts on most benchmarks.

Method	AIME24 ↑	AIME25 ↑	GSM8k ↑	MATH500 ↑	AMC23 ↑	Olympiad ↑	Minerva ↑	HMMT ↑	AVG ↑
RLOO	11.87	11.47	**91.13**	71.80	53.67	33.28	**23.16**	2.83	37.40
w/ BYPO	**15.87**	**12.80**	91.05	**73.60**	**60.33**	**37.89**	22.79	**4.05**	**39.80**
RF++	12.53	9.20	90.30	73.60	53.67	34.47	24.63	4.39	37.85
w/ BYPO	**13.07**	**10.00**	**91.05**	**74.20**	**60.33**	**35.36**	**26.47**	**4.72**	**39.40**
Dr-GRPO	**15.47**	9.87	91.28	**74.20**	**55.67**	35.51	23.90	4.94	38.85
w/ BYPO	14.13	**11.07**	**91.66**	72.20	55.50	**37.15**	**25.37**	**5.22**	**39.04**
DAPO	14.93	8.67	91.36	74.00	**59.17**	36.26	23.90	4.00	39.03
BYPO	**16.27**	**13.87**	**91.96**	**74.20**	57.67	**38.19**	**25.00**	**4.11**	**40.16**
DAPO†	13.07	10.53	**91.66**	73.60	57.50	38.48	23.53	**5.61**	39.25
w/ BYPO	**16.27**	**12.00**	90.75	**74.80**	**62.00**	**40.12**	**24.63**	5.16	**40.72**

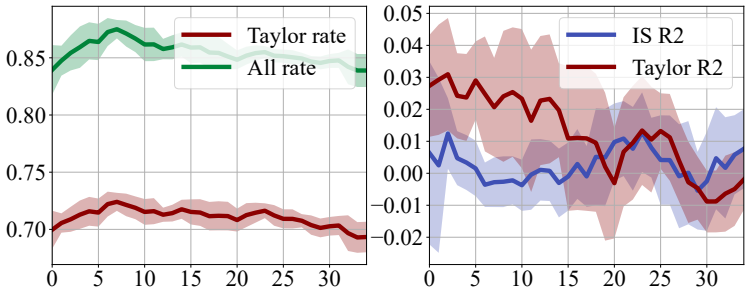


Figure 4: Activation rates of different correction terms (left) and their correlation coefficients with the correction-selection operators (right) in the Qwen2.5-7B experiments. The left panel shows that BYPO correction terms are frequently activated throughout training, while the right panel shows that their correlations with the selection operators are extremely weak, indicating that the bias introduced by the operators is negligible.

Taylor correction and only about 15% from the sequence-level importance-weight correction. This suggests that sequence-level importance weights are often excessively large, implying higher variance. The right panel shows that the correlations between the correction terms and the selection operators are consistently very small, remaining within approximately ± 0.03 and decreasing over training. Moreover, the sequence-level importance-weight correction exhibits even lower correlation, indicating that its more active selection operator introduces less additional error. Although the main text shows that these correlation coefficients characterize the bias introduced by dynamic correction selection, our results indicate that this bias is negligible in practice.

D.4 Average Benchmark Scores and Average Response-length Changes

To further understand the specific effect of BYPO, we track the average benchmark score and response length throughout training for different algorithms, as shown in Figure 5. BYPO consistently achieves the highest average benchmark accuracy and continues to improve throughout training, with even faster gains in later stages. This suggests that BYPO's error correction helps policy optimization continually identify new effective directions for improvement, enabling sustained performance growth. DAPO and Dr-GRPO also show clear continued improvement, possibly because,

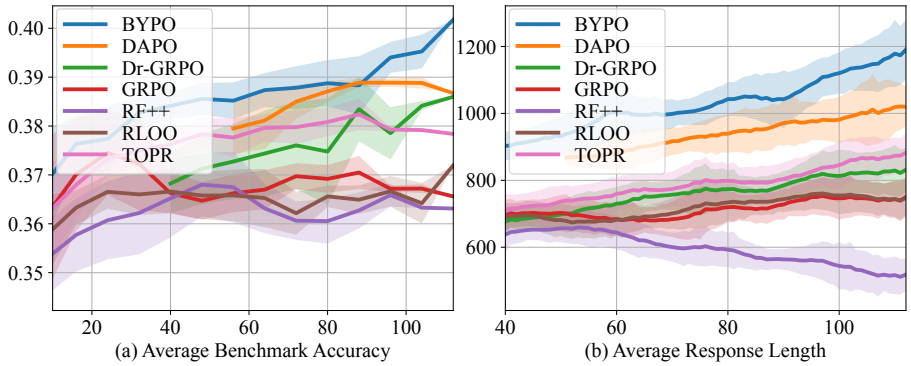


Figure 5: Average benchmark scores and average response-length changes of different algorithms in the Qwen2.5-7B experiments. BYPO consistently achieves the highest benchmark accuracy and shows a clear upward trend throughout training. DAPO and Dr-GRPO also exhibit steady improvements, but remain slightly behind BYPO. In contrast, RLOO, GRPO, and REINFORCE++ reach their peak benchmark performance relatively early and show little subsequent improvement. BYPO and DAPO also exhibit substantial increases in response length, suggesting that longer chains of thought may contribute to higher accuracy. REINFORCE++ is the only method whose average response length decreases over training, which may indicate premature entropy collapse.

as analyzed in the main text, they provide accurate first-order Taylor approximations. Most other algorithms reach their peak benchmark scores relatively early and then stagnate, suggesting that approximation error may hinder convergence and further policy improvement. The right panel shows that most algorithms steadily increase their average response length, indicating that longer reasoning processes may offer additional performance gains. In contrast, REINFORCE++ exhibits a continuous decrease in response length, which may be causally related to its lower benchmark performance.

D.5 Entropy Dynamics

To investigate how entropy evolves during training, we visualize the entropy dynamics in the Qwen2.5-7B experiments, as shown in Figure 6. We observe that the entropy of all methods decreases rapidly as training progresses. Among them, DAPO and BYPO maintain relatively higher entropy, with BYPO exhibiting the highest entropy overall. This may indicate better entropy control, suggesting that one effect of off-policy error is to drive the policy toward low-diversity regions, while BYPO provides a correction for this tendency. The right panel shows the relationship between entropy and training-set accuracy during training. We can see that nearly all methods trade entropy for accuracy, which is broadly consistent with the findings of (Cui et al., 2025). At the same time, when reaching the same level of accuracy, BYPO incurs the smallest entropy reduction among all methods. This may suggest that correcting off-policy error leads to a more efficient trade-off between diversity and accuracy.

D.6 More Baselines

To compare with a broader set of state-of-the-art RL algorithms, we additionally include three strong classical baselines, GPG (Chu et al., 2025), GSPO (Zheng et al., 2025b), and MaxRL (Tajwar et al., 2026), together with M2PO (Zheng et al., 2025c), a variance-control-based baseline, on Qwen2.5-7B. All methods are trained using the same general configuration and dataset, together with their respective default hyperparameters. The results are reported in Table 9. BYPO still achieves the highest average score and outperforms these baselines on most benchmarks.

E Case Study

To examine the advantages of BYPO in greater detail, we select responses to an AlpacaEval example from models trained with different algorithms in our RLHF experiments and compare their

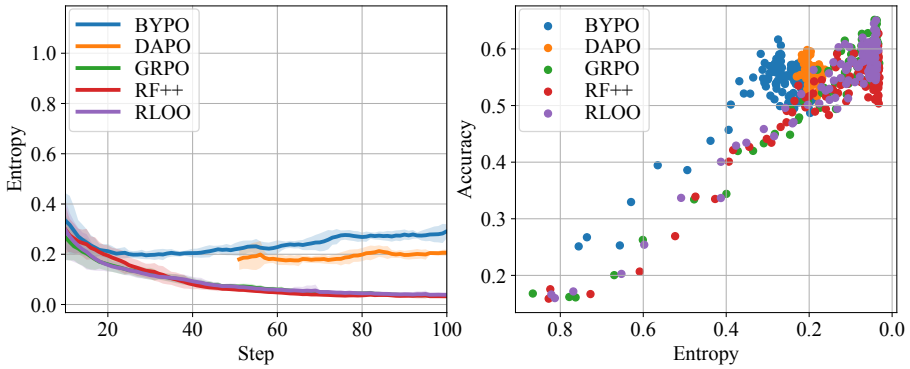


Figure 6: Entropy dynamics in the Qwen2.5-7B experiments. The left panel shows the relationship between training steps and entropy. For most algorithms, the entropy rapidly decreases to a very low level, whereas DAPO and BYPO maintain relatively higher entropy, suggesting better control over entropy collapse. The right panel shows the relationship between entropy and training-set accuracy. All methods appear to trade entropy for accuracy, but to reach the same level of accuracy, BYPO incurs the smallest reduction in entropy.

Table 9: Performance comparison between BYPO and additional baseline methods in the Qwen2.5-7B experiments. We report accuracy on standard mathematical reasoning benchmarks, as well as the average accuracy across all benchmarks (AVG). **Bold** numbers indicate the best result for each metric. BYPO achieves the highest overall score and outperforms the other methods on most benchmarks.

Method	AIME24 ↑	AIME25 ↑	GSM8k ↑	MATH500 ↑	AMC23 ↑	Olympiad ↑	Minerva ↑	HMMT ↑	AVG ↑
Base	4.00	3.33	59.06	41.40	26.00	18.22	9.19	3.11	20.54
GPG	17.07	7.20	89.76	70.80	58.00	35.22	25.37	**4.94**	38.54
GSPO	17.47	2.40	89.31	72.80	57.33	36.85	**26.10**	3.72	38.25
MaxRL	**17.73**	8.80	91.21	71.00	54.50	34.18	24.26	3.22	38.11
M2PO	12.27	12.80	91.21	73.40	**58.17**	35.81	24.26	4.72	39.08
BYPO	16.27	**13.87**	**91.96**	**74.20**	57.67	**38.19**	25.00	4.11	**40.16**

qualitative differences. As shown below, although neither response is perfect, the model trained with BYPO produces fewer factual errors overall. In contrast, the GRPO-trained model contains several serious plot and factual inaccuracies that directly undermine essay quality. For example, it claims that Romeo "attend[s] a Capulet ball without informing anyone," which supposedly leads Tybalt to challenge him on the spot and causes Mercutio's death, thereby conflating events from different scenes. It also states that Romeo marries Juliet "under the nightshade tree," which does not occur in the original play, and that Friar Lawrence advises them to wait until morning to consummate the marriage while Romeo ignores this advice, which is likewise not a reliable account of the plot. The later attribution of the Paris marriage arrangement and the communication failure primarily to Romeo is also questionable. Although the BYPO-trained model still contains imperfections, its overall line of argument is more reasonable: it focuses on Romeo's impulsiveness, his immediate decision to take poison after mistakenly believing Juliet to be dead, and his failure to wait or verify the situation when arguing that Romeo bears greater responsibility. These points better capture the key chain of actions leading to the tragedy. This suggests that correcting off-policy error with BYPO may help reduce the frequency of hallucinations in LLM outputs.

Prompt

```
Write an essay on how Romeo is most to blame for the events that
    take place in the end of the book
```

GRPO

```
While many readers might attribute the tragic ending of Shakespeare
    ’s Romeo and Juliet to fate or circumstance, a closer
    examination reveals that Romeo Montague bears significant
    responsibility for the devastating outcome. His impulsive
    decisions, hasty actions, and failure to consider consequences
    ultimately contribute to the downfall of both himself and his
    beloved Juliet.

One of the primary reasons Romeo is to blame is his reckless
    behavior when he decides to attend a Capulet ball without
    informing anyone. This decision sets off a chain reaction of
    events that leads directly to the tragedy. By attending the
    party uninvited, Romeo puts himself at risk of being discovered
    by Tybalt, a sworn enemy of the Montagues. When their paths
    cross, Tybalt challenges him to a duel, which ends with Mercutio
    ’s fatal wounding. In a fit of rage, Romeo kills Tybalt, leading
     to banishment from Verona.

Romeo’s impulsiveness also plays a crucial role in the events
    leading up to Juliet’s "death." After killing Tybalt, Romeo
    flees Verona, leaving behind his friends and family. He then
    hastily marries Juliet under the nightshade tree, demonstrating
    his lack of consideration for her well-being and future
    prospects. When Friar Lawrence advises them to wait until
    morning before consummating their marriage, Romeo ignores this
    advice and proceeds to sleep with Juliet immediately after the
    ceremony.

This haste culminates in Juliet’s forced marriage to Paris,
    arranged by her parents. Feeling trapped and desperate, Juliet
    turns to Friar Lawrence for help, who proposes the plan of using
     a potion to make it seem as though she has died. However, due
    to Romeo’s earlier rashness, they fail to communicate
    effectively about the details of the plan. As a result, Romeo
    believes Juliet is truly dead and poisons himself in her tomb,
    rather than waiting for the planned reunion.

Furthermore, Romeo’s inability to think critically and anticipate
    consequences exacerbates the situation. Throughout the play, he
    consistently prioritizes emotions over reason, often acting on
    instinct rather than careful deliberation. For example, when he
    discovers Juliet faking death, instead of taking time to
    understand the circumstances surrounding the supposed demise, he
     leaps into action, assuming the worst-case scenario.

In contrast, other characters like Friar Lawrence and even Juliet
    herself demonstrate more caution and thoughtfulness in their
    decision-making processes. While not perfect, these characters’
    attempts to mitigate the conflict between the two families are
    hindered by Romeo’s emotional recklessness.

Ultimately, Romeo’s shortsightedness and impulsivity serve as
    catalysts
```

BYPO

```
In William Shakespeare’s tragic play "Romeo and Juliet," the
    titular characters’ impulsive decisions lead to a devastating
    conclusion. While both protagonists are equally responsible for
```

```
    their fate, a closer examination reveals that Romeo's actions
    ultimately contribute more significantly to the tragic outcome.

From the beginning, Romeo's passion-driven nature sets him up for
    disaster. His hasty decision to marry Juliet without considering
     the consequences or obtaining her family's consent creates a
    precarious situation from which there can be no escape. In Act
    2, Scene 1, Romeo declares his love to Friar Lawrence, stating,
    "I have bought the mansion of a name / Where I may live with
    serenity" (II.i.91-92). This romantic idealism blindsides him to
     the reality of the feuding families and the danger it poses to
    his beloved Juliet.

Moreover, Romeo's impulsiveness leads him to act rashly when he
    discovers Juliet faking her own death. Believing she has truly
    perished, Romeo decides to poison himself in her tomb rather
    than waiting to learn if Juliet had planned this ruse all along.
     This reckless choice ensures that Juliet awakens to find her
    lover dead, sealing their demise forever. Had Romeo taken a
    moment to reflect on the potential consequences of his actions,
    he might have spared them both.

Furthermore, Romeo's failure to consider alternative solutions
    exacerbates the tragedy. Instead of seeking out help from
    friends like Mercutio or Benvolio, who could have offered
    guidance and support, Romeo isolates himself in his grief,
    making poor choices that accelerate the catastrophic finale. By
    ignoring wise counsel and acting solely on emotions, Romeo
    demonstrates a lack of maturity and foresight, leaving the
    audience to lament the avoidable tragedy.

Juliet, too, bears responsibility for her part in the drama;
    however, her mistakes are largely reactive responses to
    circumstances created by Romeo. She follows through on her plan
    to fake her own death after discovering Romeo banished,
    demonstrating her determination but also illustrating the extent
     to which she relies on Romeo's decisions. Ultimately, while
    Juliet makes errors, they stem from trying to navigate the
    complex web woven by Romeo's initial haste and subsequent
    recklessness.

In conclusion, while both Romeo and Juliet share equal culpability
    for their tragic ending, Romeo's impetuousness and inability to
    think critically about the consequences of his actions render
    him the primary catalyst for the downfall. Through his
    headstrong decisions, he carelessly exposes them both to peril,
    setting off a chain reaction that culminates in the
    heartbreaking loss of two young lives.
```

F Discussion and Additional Remarks

The main text primarily presents the key motivation and theoretical analysis of BYPO, which are sufficient for readers to understand how BYPO handles offline errors. To maintain readability, we omit detailed discussions of several aspects from the main text and provide additional clarifications here to avoid potential misunderstandings.

F.1 Necessity of Sequence-Level Importance-Weight Correction

The full BYPO method employs both sequence-level importance-weight correction and second-order Taylor-expansion correction. In the main text, we have shown that importance weighting

provides a stronger correction but incurs higher variance, whereas the second-order Taylor correction offers a trade-off between correction strength and variance. This suggests a qualitative conclusion: sequence-level importance-weight correction is more effective when the policy shift is small, whereas second-order Taylor correction is more effective when the policy shift is large. However, there is no quantitative criterion for distinguishing between small and large policy shifts, and in practice, one correction may be consistently more suitable than the other in most cases.

We observe a similar phenomenon in our experiments. As shown in Table 7 and Table 4, Setting 5, which uses only the second-order Taylor correction, is only 0.1 points lower in overall score than the full BYPO method (Setting 1), while outperforming Setting 1 on four benchmarks: MATH500, AMC23, Olympiad, and HMMT. This indicates that the performance gain brought by sequence-level importance weighting is relatively marginal. Moreover, in Table 4, only approximately 14% of the importance weights are activated, suggesting that the effect of importance weighting is also relatively limited.

Despite the limited contribution of importance weighting observed in our experiments, the full BYPO method does achieve a higher overall score than BYPO using only the second-order Taylor correction. In addition, sequence-level importance weighting is one of the predominant approaches adopted in related work. We therefore retain sequence-level importance weighting as a component of BYPO.

F.2 Magnitude of the Offline Error

In Equation 4, we show that the upper bound of the offline error is first-order in the policy shift, whereas in Section 5.2, we show that the surrogate objective (Equation 2) is in fact a first-order Taylor expansion of the RL objective (Equation 1), whose approximation error is second-order in the policy shift. In Section 5.2, we have briefly explained that this discrepancy in order arises from the use of different measures of policy shift, and we provide a more detailed explanation here.

In Equation 4, the policy shift is measured by the token-level TV divergence, whereas the error term of the Taylor expansion is characterized by the ℓ_2 norm of the policy shift over the entire action-state space, i.e., $\sqrt{\sum_{a,s,t}\left(\pi_\theta(a_t|a_{0:t},s)-\pi_{\text{old}}(a_t|a_{0:t},s)\right)^2}$. These are two different measures of policy shift, so it is natural that they yield different orders for the upper bound on the offline error. Note that the latter gives a very loose upper bound, since the policy shift over the entire action-state space is typically large, and is therefore not suitable for characterizing the magnitude of the offline error. We use it only to compare the relative approximation errors of Taylor expansions of different orders, rather than to characterize their absolute error magnitudes. For analyzing the absolute magnitude of the offline error, we recommend using the formulation in Equation 4.

F.3 Clipping in BYPO

Although BYPO incurs smaller offline error, the clipping mechanism is still retained. This is because the second-order Taylor expansion extensively used in BYPO still incurs offline error that is upper bounded by the policy shift. Although this error is smaller than that of the first-order Taylor approximation, it remains necessary to constrain it. In addition, clipping also controls the variance by limiting the policy shift. As shown in Equation 13, the variance of the objective is likewise upper bounded in terms of the policy shift. Therefore, BYPO retains the clipping mechanism as a safeguard against both offline error and variance, while using the default clipping thresholds to ensure a fair comparison with the baselines.

Clipping may appear to alter the gradient of the objective, suggesting that BYPO is no longer a strict offline-error correction. In fact, however, the BYPO objective is not an unconstrained objective obtained solely by correcting the offline error. Rather, while optimizing the corrected objective, BYPO additionally constrains the policy shift to remain within a prescribed threshold, in the same spirit as TRPO (Schulman et al., 2015), and is therefore essentially a trust-region method. As in PPO, the role of clipping is to transform this constrained optimization problem into an unconstrained objective by suppressing gradients associated with policy shifts beyond the threshold while leaving the remaining gradients unchanged.

F.4 CORRECTION-TERM SELECTION OPERATOR

BYPO selects correction terms based on their absolute magnitudes, whereas Equation 9 shows that the variance of a correction term is upper bounded by the chi-squared divergence, which provides an alternative selection criterion. We adopt the former design because Equation 9 is intended only to characterize the relative scale of the variance, and the resulting bound may be relatively loose. Directly using the chi-squared divergence as the selection criterion would eliminate too many correction terms and thereby weaken the correction effect of BYPO. In our experiments, we find that this variant underperforms the current version of BYPO by 1–2 benchmark points. Moreover, the absolute magnitude of a sample is a more direct indicator of variance, allowing high-variance samples to be filtered more precisely.