BELLMAN POLICY OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards (RLVR) has become a central approach to improving the reasoning capabilities of large language models (LLMs). We introduce Bellman Policy Optimization (BPO), a critic-free policy optimization method derived from Policy Mirror Descent (PMD). For autoregressive generation with terminal rewards, we use the Bellman equations to reformulate PMD into a trajectory-level objective that avoids estimating intermediate state values. We prove that the reformulated objective has the same unique optimal solution as the original PMD objective. By approximating the trajectory-level objective, we derive the BPO loss, which replaces the importance-sampling ratio in Group-Relative Policy Optimization (GRPO) with a smoothed ratio of complementary token probabilities. In long chain-of-thought and multi-turn tool-integrated mathematical reasoning tasks, BPO achieves the highest average accuracy over the evaluated benchmarks among all compared methods.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has become an important approach for improving the reasoning capabilities of large language models (LLMs) (Shao et al., 2024; Guo et al., 2025; Yang et al., 2025; Yu et al., 2025; Abdin et al., 2025; Wen et al., 2026). In RLVR, models are trained using outcome-level rewards assigned to generated responses by task-specific verifiers. Recent work has shown that choices in policy optimization can have a substantial impact on both reasoning performance and training stability (Yu et al., 2025; Zheng et al., 2025b; Qi et al., 2026; Ma et al., 2026a; Mao et al., 2026; Kang et al., 2026).

Group-Relative Policy Optimization (GRPO) and its variants are widely used in RLVR (Shao et al., 2024; Yu et al., 2025; Liu et al., 2025b; Zheng et al., 2025b; MiniMax et al., 2025; Zhang et al., 2026; Qi et al., 2026; Liu et al., 2025a). GRPO samples multiple responses for each prompt and computes advantages by normalizing their rewards within the group. Under outcome supervision, all tokens in a response share the same advantage. GRPO uses these advantages in a PPO-style clipped surrogate objective with token-level importance-sampling ratios (Schulman et al., 2017b). This formulation and its variants avoid training a value model and support large-scale reasoning-model training (Guo et al., 2025; Yang et al., 2025; Abdin et al., 2025; MiniMax et al., 2025; NVIDIA et al., 2025; Blakeman et al., 2026; Ling Team et al., 2025).

Motivated by its well-established theoretical foundations (Lan, 2023; Xiao, 2022; Zhan et al., 2023), we build on Policy Mirror Descent (PMD), which updates the policy using action values and a divergence penalty. However, directly applying this update to language generation requires value estimates at intermediate states. A common approach is to train a separate value model, which increases memory and computational costs (Shao et al., 2024). Learned value estimates can also be inaccurate on reasoning tasks (Kazemnejad et al., 2025). We seek a reformulation of PMD that recovers the same policy update without training a value model.

We introduce Bellman Policy Optimization (BPO), a critic-free policy optimization method derived from PMD. Our derivation builds on prior work that reparameterizes rewards and values using policy likelihood ratios (Rafailov et al., 2023; Zeng et al., 2024; Rafailov et al., 2024; Kang et al., 2026). For autoregressive generation with terminal rewards, the Bellman equations express each token advantage as a difference between value functions at consecutive states. These differences telescope along a response, leaving only the terminal reward and the initial value. We combine this identity with the PMD optimality condition to derive a trajectory-level objective. The initial value is

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.

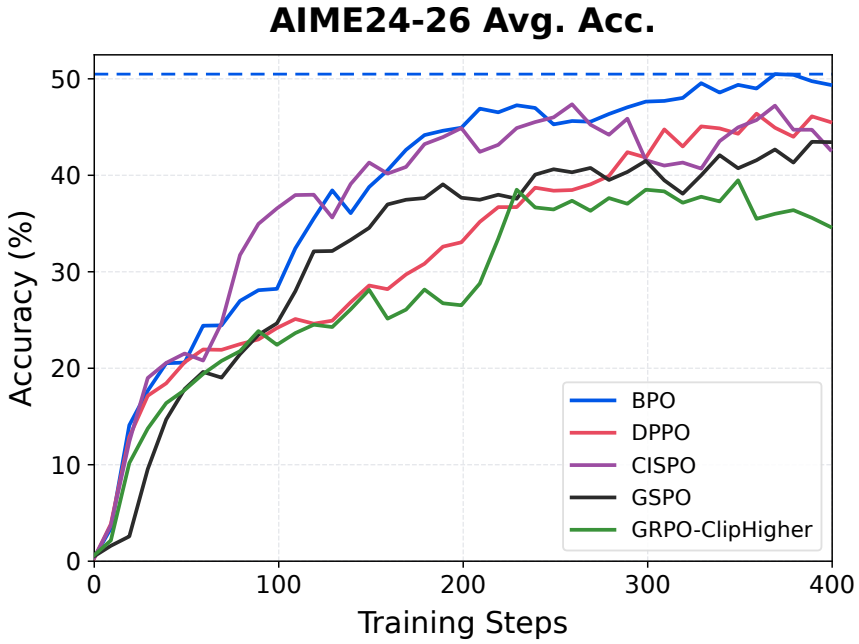

Figure 1: Evaluation accuracy during training on Qwen3-30B-A3B-Base, comparing BPO with GRPO-ClipHigher, GSPO, CISPO, and DPPO. All methods are trained for 400 training steps on the English subset of DAPO-Math-17k under identical experimental settings except for the policy loss. Each rollout batch contains 256 prompts, with a group size of 16 responses per prompt and a maximum response length of 16384 tokens. One training step consists of one rollout batch followed by eight optimizer updates. Curves show mean Avg@32 accuracy across AIME24–26, with 32 responses sampled per question to estimate Pass@1. The horizontal dashed line marks BPO's peak mean accuracy.

the expected reward for a prompt and can be estimated from sampled responses. The reformulation therefore avoids estimating values or advantages at intermediate states. We prove that this objective and the original PMD objective have the same unique optimal solution on states reachable under the rollout policy.

We then derive the practical BPO loss through a sequence of approximations. We estimate the initial value using the mean reward of responses sampled for each prompt. We approximate the gradient of the trajectory-level objective and replace the full KL divergence with a binary approximation (Qi et al., 2026). The resulting gradient includes a mismatch-correction weight determined by the rollout and current token probabilities. We apply additive smoothing to this weight for numerical stability. In the practical BPO loss, this weight replaces the importance-sampling ratio used in GRPO.

We evaluate BPO on long chain-of-thought (CoT) mathematical reasoning with Qwen3-30B-A3B-Base trained on DAPO-Math-17k. We compare against GRPO-ClipHigher (Shao et al., 2024; Yu et al., 2025), GSPO (Zheng et al., 2025b), CISPO (MiniMax et al., 2025), and DPPO (Qi et al., 2026) under identical experimental settings. As shown in Figure 1, BPO achieves a peak average accuracy of 50.5% across AIME24–26 (Art of Problem Solving, 2026). The gains over these four baselines are 11.0, 7.0, 3.1, and 4.1 percentage points, respectively. At the final checkpoint, BPO achieves 45.0% average accuracy over six benchmarks from AMC, AIME, and HMMT, 2.7–9.7 percentage points higher than the four baselines, and achieves the best score on every benchmark. In multi-turn tool-integrated reasoning with Qwen3-4B-Instruct-2507, we compare against GRPO-ClipHigher, CISPO, and DPPO. BPO again improves over these baselines in average accuracy and achieves the best score on every benchmark.

Our main contributions are:

• We derive a critic-free, trajectory-level reformulation of PMD for autoregressive generation with terminal rewards, avoiding value or advantage estimation at intermediate states. We prove that

this reformulation and the original PMD objective have the same unique optimal solution on states reachable under the rollout policy.

• We derive the practical BPO loss from this reformulation through group-based estimation and gradient approximations. The loss replaces GRPO's importance-sampling ratio with a mismatch-correction weight given by a smoothed ratio of complementary token probabilities.

• We evaluate BPO on long-CoT mathematical reasoning and multi-turn tool-integrated reasoning. In both settings, BPO outperforms all baselines and achieves the best score on every benchmark at the final checkpoint.

2 PRELIMINARIES

2.1 RLVR AS AN EPISODIC MDP

RLVR for LLMs can be formulated as a finite-horizon episodic Markov Decision Process (MDP).

Prompts and responses. Let $\mathcal{V}$ denote the vocabulary, and let $\mathcal{D}$ denote a prompt distribution on a prompt set D. We assume $\mathcal{D}$ is uniform on D for simplicity; the analysis extends directly to more general prompt distributions. A prompt $x \in D$ and a response $y = (y_1, \ldots, y_T)$ are token sequences over $\mathcal{V}$, where the length T, also denoted $|y|$, may vary across responses. A response terminates when the EOS token is generated or the response length limit $T_{\max}$ is reached.

States and transitions. The state at step t is $s_t = (x, y_{<t})$, the concatenation of the prompt x and the prefix $y_{<t} = (y_1, \ldots, y_{t-1})$. Since $y_{<1}$ is empty, the initial state is $s_1 = x$. Let $\mathcal{S}$ denote the state space. The action at step t is the token y_t, and the transition is deterministic: $s_{t+1} = (s_t, y_t)$. The terminal state of a complete response y is $s_{|y|+1} = (x, y)$.

Policies and completion distributions. A stochastic policy $\pi : \mathcal{S} \to \Delta(\mathcal{V})$ maps each state s_t to a next-token distribution $\pi(\cdot|s_t)$, which we also write as $\pi(\cdot|x, y_{<t})$. Here, $\Delta(\mathcal{V})$ is the set of probability distributions on $\mathcal{V}$. In practice, π is parameterized by an autoregressive LLM. Let $\Pi = \Delta(\mathcal{V})^{\mathcal{S}}$ denote the policy space, and let $\Pi_\pi \subseteq \Pi$ denote the set of policies that are absolutely continuous with respect to π. For a state s_t, the completion distribution of $y_{\geq t} = (y_t, \ldots, y_T)$ induced by π is $\mathbb{P}_\pi(y_{\geq t}|s_t) = \prod_{u=t}^{T} \pi(y_u|s_u)$.

Rewards and objective. A deterministic verifier assigns the terminal reward $R(x, y)$, where we assume $|R(x, y)| \leq 1$ without loss of generality. The policy objective is $\mathcal{J}(\pi) = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\mathbb{P}_\pi(\cdot|x)}[R(x, y)]$.

2.2 IMPORTANCE-SAMPLING-BASED POLICY GRADIENT METHODS

In practical RLVR, responses are generated by a rollout policy μ, while policy optimization is performed on the current policy π. Policy updates between rollout generation and training steps, together with numerical discrepancies between the rollout and training engines, can cause the two policies to differ (Zheng et al., 2025a; Qi et al., 2025; 2026; Yao et al., 2026; 2025). For each prompt $x \sim \mathcal{D}$, we independently sample a group of G responses $\{y^i\}_{i=1}^{G}$ from $\mathbb{P}_\mu(\cdot|x)$, and let $s_t^i = (x, y_{<t}^i)$ denote the state at step t of response y^i. Following Qi et al. (2026, Appendix C.1), the gradients of importance-sampling (IS)-based policy gradient methods can be written in the unified form

$$\nabla\mathcal{L}(\pi) = -\mathbb{E}_{x\sim\mathcal{D},\, \{y^i\}_{i=1}^{G}\sim\mathbb{P}_\mu(\cdot|x)}\left[\frac{1}{G}\sum_{i=1}^{G}\sum_{t=1}^{|y^i|} M_t^i \min\left\{r_t^i, C\right\}\hat{A}^i \nabla \log \pi(y_t^i|s_t^i)\right], \tag{1}$$

where r_t^i is the token-level IS ratio

$$r_t^i = \frac{\pi(y_t^i|s_t^i)}{\mu(y_t^i|s_t^i)}. \tag{2}$$

$\hat{A}^i$ is the group-normalized advantage of response y^i defined in (32), $M_t^i \in \{0, 1\}$ is a token mask, and C caps the IS ratio. The ratio r_t^i corrects for sampling from μ instead of π.

As shown in Qi et al. (2026, Appendix C.1), GRPO (Shao et al., 2024), CISPO (MiniMax et al., 2025), and DPPO (Qi et al., 2026) are instances of (1) that differ only in M_t^i and C, and GSPO (Zheng et al., 2025b) extends importance sampling to the sequence level. Appendix D gives their definitions.

2.3 Bellman Equations with Terminal Rewards

For a policy π and state $s_t = (x, y_{<t})$, the value function is the expected reward of completions under π (Sutton & Barto, 2018):

$$V^\pi(s_t) = \mathbb{E}_{y_{\geq t} \sim \mathbb{P}_\pi(\cdot|s_t)} \left[R(x, y) \right], \quad \text{where } y = (y_{<t}, y_{\geq t}). \tag{3}$$

In particular, $V^\pi(s_1) = V^\pi(x)$ is the expected reward for prompt x. Since intermediate rewards are zero and transitions are deterministic, for a non-terminal state s_t, the action-value and advantage functions are

$$Q^\pi(s_t, y_t) = V^\pi(s_{t+1}), \tag{4}$$

$$A^\pi(s_t, y_t) = Q^\pi(s_t, y_t) - V^\pi(s_t). \tag{5}$$

For any non-terminal state s_t, the Bellman equation is

$$V^\pi(s_t) = \sum_{y_t' \in \mathcal{V}} \pi(y_t'|s_t) Q^\pi(s_t, y_t') = \sum_{s' = (s_t, y_t'): y_t' \in \mathcal{V}} \pi(y_t'|s_t) V^\pi(s'),$$

with the terminal condition $V^\pi(s_{|y|+1}) = R(x, y)$ for a complete response y. It follows from (3) and (5) that

$$\sum_{y_t \in \mathcal{V}} \pi(y_t|s_t) A^\pi(s_t, y_t) = 0. \tag{6}$$

2.4 Policy Mirror Descent

Let $\mathcal{S}$ and $\mathcal{V}$ denote the state space and the vocabulary defined in Section 2.1. Given a real-valued function $f : \mathcal{S} \times \mathcal{V} \to \mathbb{R}$, Policy Mirror Descent (PMD) (Geist et al., 2019; Lan, 2023; Xiao, 2022; Tomar et al., 2022) solves the following optimization problem for each state $s \in \mathcal{S}$:

$$\max_{\pi(\cdot|s) \in \Delta(\mathcal{V})} \quad \mathbb{E}_{y \sim \pi(\cdot|s)} \left[f(s, y) \right] - \frac{1}{\eta} D_{\mathrm{KL}}(\pi(\cdot|s) \,\|\, \mu(\cdot|s)), \tag{7}$$

where μ is the behavior policy (rollout policy) and $\eta > 0$ is the step size.

The optimization problem (7) has the unique solution

$$\pi^+_{\mu,f}(y|s) = \frac{\mu(y|s) \exp\left(\eta f(s, y)\right)}{Z_{\mu,f}(s)}, \tag{8}$$

where the partition function is $Z_{\mu,f}(s) = \mathbb{E}_{y' \sim \mu(\cdot|s)} \exp\left(\eta f(s, y')\right)$.

2.5 Binary KL Divergence

Computing the full KL divergence $D_{\mathrm{KL}}(\pi(\cdot|s) \,\|\, \mu(\cdot|s))$ requires logits for the entire vocabulary and can be costly. Qi et al. (2026) propose the binary approximation. For a given action $y \in \mathcal{V}$, it partitions the action space into $\{y\}$ and its complement $\mathcal{V} \setminus \{y\}$. The binary KL divergence $D_{\mathrm{KL}}^{\mathrm{bin}}\left(\pi(\cdot|s) \,\|\, \mu(\cdot|s); y\right)$ is the KL divergence between the induced Bernoulli distributions of $\pi(\cdot|s)$ and $\mu(\cdot|s)$. Appendix B gives its explicit form in (17).

3 Bellman Policy Optimization

We first define the BPO loss and then derive it from PMD.

BPO Loss Definition. For each prompt $x \sim \mathcal{D}$, we independently sample a group of G responses $\{y^i\}_{i=1}^G$. For the t-th token of response y^i, the BPO loss is

$$\mathcal{L}^{\text{BPO}}(\pi) = -\hat{A}^i M_t^i \min\left\{\texttt{sg}\left(\omega_t^i\right), C\right\} \log \pi(y_t^i | x, y_{<t}^i), \tag{9}$$

where $\hat{A}^i$ is the advantage of response y^i in the group $\{y^i\}_{i=1}^G$ defined in (32), and ω_t^i is the mismatch-correction weight for the t-th token in response y^i, defined as

$$\omega_t^i = \frac{1 + \epsilon - \mu(y_t^i | x, y_{<t}^i)}{1 + \epsilon - \pi(y_t^i | x, y_{<t}^i)}. \tag{10}$$

The constant C caps ω_t^i to stabilize training, and the mask M_t^i follows (30) with ω_t^i in place of r_t^i (Appendix E). The BPO per-token loss retains the GRPO form in (29) and replaces the importance-sampling ratio r_t^i with the truncated mismatch-correction weight $\min\left\{\texttt{sg}\left(\omega_t^i\right), C\right\}$.

Derivation of the BPO Objective. We derive the BPO objective (9) in three steps:

- Step 1 (Starting Point, (19)): we instantiate PMD with $f(s, y) = A^\mu(s, y)$, where $A^\mu(s, y)$ is the advantage function introduced in Section 2.3.
- Step 2 (Critic-free PMD, Section 3.1): we derive a critic-free objective with the same optimal solution as advantage-based PMD. This avoids estimating $A^\mu(s, y)$ at individual states.
- Step 3 (Practical Approximation, Section 3.2): we approximate the critic-free PMD objective from Step 2 to obtain the BPO objective (9).

Starting Point: Advantage-Based PMD. We first instantiate the general PMD objective in (7) with the rollout-policy advantage of the terminal-reward MDP in Section 2.3, i.e., $f(s, y) = A^\mu(s, y)$, and refer to the resulting problem as advantage-based PMD. Classical PMD is commonly formulated using the action-value function $Q^\mu(s_t, y_t)$ (Geist et al., 2019; Lan, 2023; Xiao, 2022; Tomar et al., 2022). Using A^μ gives the same update because $Q^\mu(s_t, y_t) = A^\mu(s_t, y_t) + V^\mu(s_t)$, and the term $V^\mu(s_t)$ does not depend on the action y_t. By (8), advantage-based PMD has the unique solution $\pi^+ = \pi^+_{\mu, A^\mu}$, with partition function $Z_\mu = Z_{\mu, A^\mu}$. Appendix C.1 writes out the problem (19) and its solution (20) at each state $s_t = (x, y_{<t})$.

3.1 Critic-Free Reformulation of Policy Mirror Descent

Directly implementing the update (19) requires estimating $A^\mu(s_t, y_t)$ at every visited intermediate state. This typically requires a learned critic or costly conditional rollouts. We derive the critic-free reformulation in (11) by expressing rewards and values through policy likelihood ratios and the Bellman equations. This follows prior work on direct alignment (Rafailov et al., 2023; Zeng et al., 2024; Rafailov et al., 2024; Kang et al., 2026).

Critic-Free Reformulation of PMD. Let μ denote the rollout policy. We assign each prompt $x \in D$ a fixed positive weight $\phi(x)$. We will specify this weight in Section 3.2 to obtain the practical BPO loss. We consider the following critic-free objective:

$$\min_{\pi \in \Pi_\mu} \mathcal{L}(\pi) = \mathbb{E}_{x \sim \mathcal{D}, y \sim \mathbb{P}_\mu(\cdot|x)} \left[\phi(x) \cdot \frac{\delta(x, y; \pi, \mu)^2}{2\eta} \right], \tag{11}$$

where the trajectory-level residual $\delta(x, y; \pi, \mu)$ is defined as

$$\delta(x, y; \pi, \mu) = \eta\big(R(x, y) - V^\mu(x)\big) - \sum_{t=1}^{|y|} \Big(\log \frac{\pi(y_t|s_t)}{\mu(y_t|s_t)} + D_{\text{KL}}(\mu(\cdot|s_t) \,\|\, \pi(\cdot|s_t)) \Big). \tag{12}$$

The reformulation (11) does not depend on state-value functions at intermediate states. Instead, it only requires the terminal reward $R(x, y)$ and the initial value function $V^\mu(x)$. The terminal reward $R(x, y)$ is directly provided by the verifier, while $V^\mu(x) = \mathbb{E}_{y \sim \mathbb{P}_\mu(\cdot|x)}[R(x, y)]$ is the expected reward for prompt x under the rollout policy μ. We can estimate this expectation from rollout data.

Equivalence to PMD. Theorem 1 shows that the critic-free objective (11) and PMD (19) have the same unique optimal solution. This equivalence holds for any positive weighting function ϕ. We specify ϕ in Section 3.2 when deriving the practical BPO loss (9).

Theorem 1. *Let the policy π^+ in* (20) *denote the optimal solution of the PMD problem* (19)*. Let $\phi : D \to \mathbb{R}_+$ be any positive weight function on the prompt set D. For any prompt $x \in D$, we solve the optimization problem* (11)*. Then, this optimization problem admits a unique optimal solution $\hat{\pi}^+$ which equals π^+. Here, uniqueness and equivalence are in the sense that any policy $\hat{\pi}^+ \in \Pi_\mu$ that solves* (11) *induces the same completion distribution as π^+ for every prompt $x \in D$, i.e., $\mathbb{P}_{\hat{\pi}^+}(\cdot|x) = \mathbb{P}_{\pi^+}(\cdot|x)$.*

Appendix C.2 derives objective (11) from the advantage-based PMD problem (19), and Appendix C.3 proves Theorem 1.

3.2 Practical Approximation

We obtain the practical BPO loss (9) by approximating the critic-free objective (11). We linearize the squared-residual objective and estimate the initial value and prompt-dependent scaling factor from grouped rollouts. We then approximate the full reverse KL divergence using binary KL divergence and apply smoothing, masking, and clipping.

For response y^i, define the response-level critic-free loss $\mathcal{L}_i(\pi) = \phi(x)\delta(x, y^i; \pi, \mu)^2/(2\eta)$ as in (11). Its gradient can be decomposed into token-level contributions $\nabla\mathcal{L}_i(\pi) = \sum_{t=1}^{|y^i|} \nabla\mathcal{L}_{i,t}(\pi)$. Throughout this section, ∇ denotes the gradient with respect to the parameters of π with μ fixed.

• Step 1: linearized approximation. We linearize the squared-residual loss with respect to the residual δ, around its value at $\pi = \mu$. Since $\delta(x, y^i; \mu, \mu) = \eta\left(R(x, y^i) - V^\mu(x)\right)$, this replaces the residual factor $\delta(x, y^i; \pi, \mu)/\eta$ in the gradient by $R(x, y^i) - V^\mu(x)$. For response y^i and the corresponding state $s_t^i = (x, y^i_{<t})$, the per-token gradient contribution is approximated by

$$\begin{aligned}\nabla\mathcal{L}_{i,t}(\pi) &= -\phi(x)\frac{\delta(x, y^i; \pi, \mu)}{\eta}\nabla\Big(\log\pi(y_t^i|s_t^i) + D_{\mathrm{KL}}(\mu(\cdot|s_t^i)\,\|\,\pi(\cdot|s_t^i))\Big) \\ &\approx -\phi(x)\Big(R(x, y^i) - V^\mu(x)\Big)\nabla\Big(\log\pi(y_t^i|s_t^i) + D_{\mathrm{KL}}(\mu(\cdot|s_t^i)\,\|\,\pi(\cdot|s_t^i))\Big).\end{aligned} \tag{13}$$

• Step 2: group-based estimation and normalization. The initial value function $V^\mu(x) = \mathbb{E}_{y\sim\mathbb{P}_\mu(\cdot|x)}\left[R(x, y)\right]$ has the unbiased estimator $\texttt{mean}\big(\{R(x, y^j)\}_{j=1}^G\big)$. We further choose $\phi(x)$ as the inverse reward standard deviation, i.e., $\phi(x) = \frac{1}{\sqrt{\mathrm{Var}_{y\sim\mathbb{P}_\mu(\cdot|x)}(R(x,y))}}$. We replace it by its empirical counterpart $\phi(x) \approx \frac{1}{\texttt{std}\big(\{R(x,y^i)\}_{i=1}^G\big)}$. With these substitutions, (13) becomes

$$\nabla\mathcal{L}_{i,t}(\pi) \approx -\hat{A}^i\nabla\Big(\log\pi(y_t^i|s_t^i) + D_{\mathrm{KL}}(\mu(\cdot|s_t^i)\,\|\,\pi(\cdot|s_t^i))\Big). \tag{14}$$

• Step 3: binary KL approximation and additive smoothing. Computing the full reverse KL divergence in (14) can be costly. We therefore replace the full KL divergence $D_{\mathrm{KL}}(\mu(\cdot|s_t^i)\,\|\,\pi(\cdot|s_t^i))$ in (14) with the binary KL divergence $D_{\mathrm{KL}}^{\mathrm{bin}}\left(\mu(\cdot|s_t^i)\,\|\,\pi(\cdot|s_t^i); y_t^i\right)$ as defined in Section 2.5. As proved in Appendix C.4, the following identity holds

$$\nabla\Big(\log\pi(y_t^i|s_t^i) + D_{\mathrm{KL}}^{\mathrm{bin}}\left(\mu(\cdot|s_t^i)\,\|\,\pi(\cdot|s_t^i); y_t^i\right)\Big) = \frac{1-\mu(y_t^i|s_t^i)}{1-\pi(y_t^i|s_t^i)}\nabla\log\pi(y_t^i|s_t^i). \tag{15}$$

Substituting (15) into (14) and replacing the multiplier $\frac{1-\mu(y_t^i|s_t^i)}{1-\pi(y_t^i|s_t^i)}$, which can become large when $\pi(y_t^i|s_t^i)$ approaches one, with the additively smoothed mismatch-correction weight ω_t^i in (10) for numerical stability gives

$$\nabla\mathcal{L}_{i,t}(\pi) \approx -\hat{A}^i\frac{1-\mu(y_t^i|s_t^i)}{1-\pi(y_t^i|s_t^i)}\nabla\log\pi(y_t^i|s_t^i) \approx -\hat{A}^i\omega_t^i\nabla\log\pi(y_t^i|s_t^i). \tag{16}$$

• Step 4: masking and clipping. Applying the GRPO-style clipping mask M_t^i and capping ω_t^i at C in (16) yields $-\hat{A}^i M_t^i \min\left\{\omega_t^i, C\right\}\nabla\log\pi(y_t^i|s_t^i)$, which is the gradient of the BPO loss (9).

Table 1: Avg@32 (%) on Qwen3-30B-A3B-Base at the final checkpoint. The best result in each column is bold and the second best is underlined.

METHOD	AMC23	AIME24	AIME25	AIME26	HMMT25	HMMT26	AVG.
GRPO-ClipHigher	87.9	40.6	31.1	32.2	9.7	10.0	35.3
GSPO	<u>93.5</u>	49.9	37.0	43.4	<u>15.4</u>	8.4	41.3
CISPO	88.4	47.7	36.1	<u>44.1</u>	14.4	9.2	40.0
DPPO	93.2	<u>55.0</u>	<u>37.6</u>	44.0	13.7	<u>10.3</u>	<u>42.3</u>
BPO	**95.2**	**57.8**	**40.9**	**49.4**	**16.0**	**10.4**	**45.0**

4 EXPERIMENTS

We evaluate BPO in long chain-of-thought (CoT) mathematical reasoning and multi-turn tool-integrated reasoning. In each setting, we vary only the policy loss and keep all other experimental settings identical across methods. All methods are trained on the English subset of DAPO-Math-17k (Yu et al., 2025). Tables 3 and 4 in Appendix F list the hyperparameters.

We evaluate on AMC23 (Li et al., 2024), AIME24, AIME25, AIME26 (Art of Problem Solving, 2026), HMMT25, and HMMT26 (Balunović et al., 2025). We estimate Pass@1 using Avg@32 In the tables, the average accuracy (Avg.) is the arithmetic mean over the six benchmarks. During training, we evaluate on AIME24–26 every 10 training steps, and the training curves show the mean accuracy over AIME24–26.

4.1 MAIN RESULTS

Long-CoT Mathematical Reasoning. We train Qwen3-30B-A3B-Base (Yang et al., 2025) and compare BPO with GRPO-ClipHigher (Shao et al., 2024; Yu et al., 2025), GSPO (Zheng et al., 2025b), CISPO (MiniMax et al., 2025), and DPPO (Qi et al., 2026). Each rollout batch contains 256 prompts with 16 responses per prompt. The resulting 4096 responses are split into 8 minibatches of 512 responses. One training step consists of one rollout batch followed by its eight minibatch updates. We train each method for 400 training steps, corresponding to 3200 optimizer updates. The maximum response length is 16384 tokens. All methods use Rollout Routing Replay (R3) (Ma et al., 2026b). BPO uses $\epsilon = 0.1$ and $C = 3.0$. Ablations on these hyperparameters and on the design of BPO are presented in Appendix H.

Table 1 reports results at the final checkpoint. BPO achieves 45.0% average accuracy. This is 2.7 percentage points higher than DPPO, the strongest baseline, and 9.7 points higher than GRPO-ClipHigher. BPO achieves the best score on all six benchmarks. As shown in Figure 1, BPO reaches the highest peak mean accuracy over AIME24–26, 50.5%, which is 3.1–11.0 percentage points higher than the peaks of the baselines, and remains at 49.4% at the final checkpoint. Results at the checkpoint with the highest mean accuracy are given in Appendix G.1, where BPO also achieves the highest average accuracy.

Multi-Turn Tool-Integrated Reasoning. We follow the tool-integrated reasoning setup of ReTool (Feng et al., 2026). Over multiple turns, the model interleaves natural-language reasoning with Python code, and a code interpreter executes the code and returns its output to the model. We start from Qwen3-4B-Instruct-2507 (Yang et al., 2025) after cold-start supervised fine-tuning on the ReTool-SFT dataset released by Feng et al. (2026). Each rollout batch contains 128 prompts with 16 responses per prompt, split into 4 minibatches. The maximum response length is 16384 tokens. We train each method for 300 training steps. We compare BPO with GRPO-ClipHigher, CISPO, and DPPO. CISPO and DPPO are the strongest baselines in Section 4.1 at the best checkpoint and at the final checkpoint, respectively.

Table 2 reports results at the final checkpoint. BPO achieves the highest average accuracy of 65.1%, compared with 63.9% for CISPO and 61.8% for GRPO-ClipHigher. BPO achieves the best score on all six benchmarks. As shown in Figure 2, BPO also reaches the highest peak mean accuracy

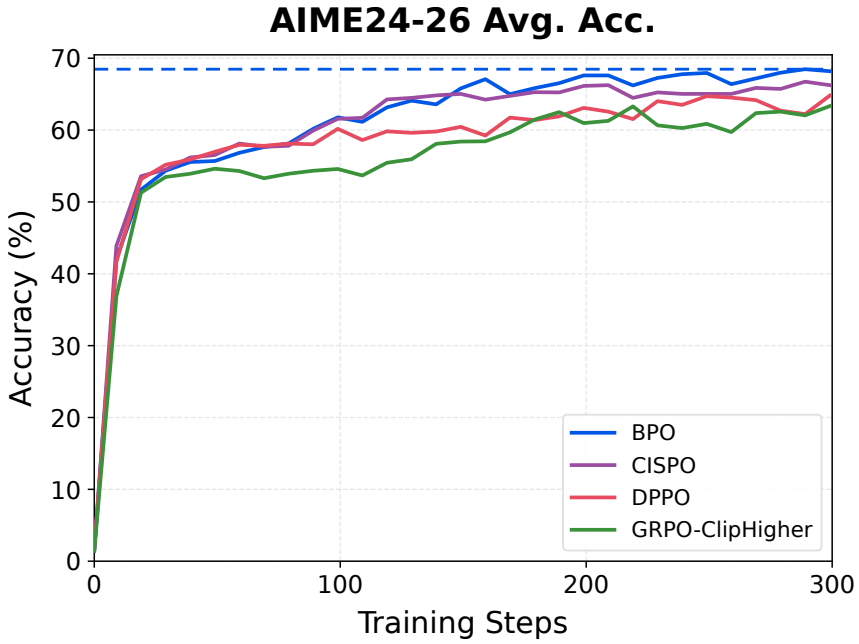

Figure 2: Evaluation accuracy during multi-turn tool-integrated training on Qwen3-4B-Instruct-2507 with ReTool cold-start fine-tuning. Curves show mean Avg@32 accuracy across AIME24–26. The horizontal dashed line marks BPO's peak mean accuracy.

Table 2: Avg@32 (%) of multi-turn tool-integrated reasoning on Qwen3-4B-Instruct-2507 at the final checkpoint. The best result in each column is bold and the second best is underlined.

METHOD	AMC23	AIME24	AIME25	AIME26	HMMT25	HMMT26	AVG.
GRPO-ClipHigher	95.2	65.7	60.2	64.1	41.0	44.4	61.8
CISPO	<u>96.8</u>	<u>69.3</u>	<u>62.6</u>	66.9	<u>42.7</u>	<u>44.9</u>	<u>63.9</u>
DPPO	96.1	64.3	61.7	<u>68.4</u>	42.3	43.3	62.7
BPO	**97.5**	**70.4**	**65.0**	**69.2**	**43.1**	**45.2**	**65.1**

over AIME24–26, 68.5%, which is 1.8–5.2 percentage points higher than the peaks of the baselines. Results at the best checkpoint are given in Appendix G.1.

4.2 Training Dynamics

Figure 3 plots the mismatch-correction weight against the rollout probability μ in the long-CoT experiment of Section 4.1. For GRPO-ClipHigher, the weight is the importance-sampling ratio r_t^i in (2). For BPO, it is ω_t^i in (10). At a training step, we collect all tokens from the 256×16 responses of that step and uniformly subsample 10^5 tokens. The current-policy probability π is taken when the token is used for training, over all minibatch updates of the step. Figure 3 shows training step 299. Appendix G.2 shows training steps 99, 199, 299, and 399, together with the corresponding density plots.

The GRPO-ClipHigher ratios form a triangle. Their spread is largest for low-probability tokens and shrinks as μ approaches one. In contrast, the BPO weights stay close to one for both low and high values of μ, and spread more evenly in between. A change in the probability of a low-probability token changes ω_t^i only slightly, while the same change can move r_t^i far from one. This pattern holds at all four training steps (Appendix G.2).

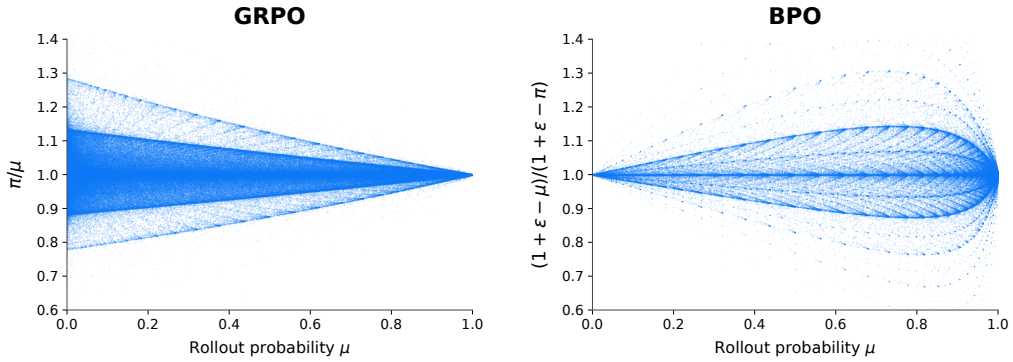

Figure 3: Mismatch-correction weight versus rollout probability μ on Qwen3-30B-A3B-Base at training step 299. Left: importance-sampling ratio π/μ of GRPO-ClipHigher. Right: BPO weight $(1+\epsilon-\mu)/(1+\epsilon-\pi)$. Each panel shows 10^5 tokens uniformly subsampled from the rollout batch of that step. Results at training steps 99, 199, 299, and 399 are given in Appendix G.2.

5 Related Work

Reinforcement learning with verifiable rewards. RLVR has become an important approach for improving the reasoning capabilities of LLMs (Lambert et al., 2025; Shao et al., 2024; Guo et al., 2025; Yang et al., 2025). Critic-free methods have become the dominant choice in RLVR (Ahmadian et al., 2024; Shao et al., 2024; Hu et al., 2025; Yu et al., 2025; Zheng et al., 2025b; MiniMax et al., 2025; Zhang et al., 2026). BPO is also critic-free and derives its per-token weight from PMD.

Policy mirror descent. PMD updates the policy at every state by maximizing the action value minus a Bregman divergence to the current policy (Geist et al., 2019; Tomar et al., 2022; Lan, 2023). For LLMs, several methods implement a mirror-descent step by regressing policy log-ratios onto rewards, with the KL penalty placed on complete responses (Gao et al., 2024; Wu et al., 2025; Zhang et al., 2025; Kimi Team, 2025; Richemond et al., 2024; Brantley et al., 2025). BPO starts instead from the state-wise update (19), which reweights each next-token distribution of μ by the advantage A^{μ}. Appendix A discusses related work in more detail, including training-inference mismatch, Bellman equations, and trust regions.

6 Conclusion

We introduced Bellman Policy Optimization (BPO), a critic-free method for RLVR derived from Policy Mirror Descent. Using the Bellman equations, we obtained a trajectory-level objective that avoids estimating intermediate state values. We proved that this objective and the original PMD objective have the same unique optimal solution on states reachable under the rollout policy. We then derived a practical token-level loss through approximations, with a smoothed mismatch-correction weight replacing the importance-sampling ratio. In both the long-CoT and the multi-turn tool-integrated settings, BPO outperforms all compared methods and achieves the highest average accuracy over the evaluated benchmarks at the final checkpoint.

AI use statement

We used generative AI tools to edit and proofread the language of this manuscript and to assist with debugging code. We have reviewed all AI-assisted work, including all AI-polished text and all code modified with AI assistance. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Reproducibility statement

The BPO loss is defined in (9), with the mismatch-correction weight in (10) and the clipping mask in Appendix E; Section 3.2 lists every approximation used to obtain this loss from the critic-free objective (11). The assumptions of Theorem 1 are stated in Section 2.1 and in the theorem itself. Appendix C.2 derives (11), and Appendices C.3 and C.4 give the complete proofs. The binary and top-k KL divergences are defined in Appendix B, and the compared methods GRPO-ClipHigher, GSPO, CISPO, and DPPO are specified in Appendix D. Section 4 describes the models, training data, benchmarks, and evaluation protocol. Appendix F gives the advantage (32), all hyperparameters (Tables 3 and 4), the hyperparameters of each baseline, the sources of the datasets and benchmarks, and the rule for selecting the best checkpoint. Appendix H describes the setup of each ablation.

References

Marah Abdin, Sahaj Agarwal, Ahmed Awadallah, Vidhisha Balachandran, Harkirat Behl, Lingjiao Chen, Gustavo de Rosa, Suriya Gunasekar, Mojan Javaheripi, Neel Joshi, et al. Phi-4-reasoning technical report. *arXiv preprint arXiv:2504.21318*, 2025.

Alekh Agarwal, Sham M. Kakade, Jason D. Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *Journal of Machine Learning Research*, 22(98):1–76, 2021.

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.

Art of Problem Solving. AIME problems and solutions, 2026. URL `https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions`.

Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. MathArena: Evaluating LLMs on uncontaminated math competitions. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, volume 38, 2025.

Aaron Blakeman, Aaron Thomas, Aastha Jhunjhunwala, Abhibha Gupta, Abhinav Khattar, Adam Rajfer, Adi Renduchintala, Adil Asif, Aditya Vavre, Adriana Flores Miranda, et al. Nemotron 3 Ultra: Open, efficient mixture-of-experts hybrid Mamba-transformer model for agentic reasoning. *arXiv preprint arXiv:2606.15007*, 2026.

Kianté Brantley, Mingyu Chen, Zhaolin Gao, Jason D. Lee, Wen Sun, Wenhao Zhan, and Xuezhou Zhang. Accelerating RL for LLM reasoning with optimal advantage regression. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Shicong Cen, Chen Cheng, Yuxin Chen, Yuting Wei, and Yuejie Chi. Fast global convergence of natural policy gradient methods with entropy regularization. *Operations Research*, 70(4):2563–2578, 2022.

Pierre Clavier, Nathan Grinsztajn, Raphaël Avalos, Yannis Flet-Berliac, Irem Ergun, Omar Darwiche Domingues, Olivier Pietquin, Pierre Harvey Richemond, Florian Strub, and Matthieu Geist. ShiQ: Bringing back Bellman to LLMs. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

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

Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. ReTool: Reinforcement learning for strategic tool use in LLMs. In *International Conference on Learning Representations*, 2026.

Wei Fu, Jiaxuan Gao, Xujie Shen, Chen Zhu, Zhiyu Mei, Chuyi He, Shusheng Xu, Guo Wei, Jun Mei, Jiashu Wang, Tongkai Yang, Binhang Yuan, and Yi Wu. AReaL: A large-scale asynchronous reinforcement learning system for language reasoning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 36256–36282, 2025.

Zhaolin Gao, Jonathan D. Chang, Wenhao Zhan, Owen Oertell, Gokul Swamy, Kianté Brantley, Thorsten Joachims, J. Andrew Bagnell, Jason D. Lee, and Wen Sun. REBEL: Reinforcement learning via regressing relative rewards. In *Advances in Neural Information Processing Systems*, volume 37, pp. 52354–52400, 2024.

Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized Markov decision processes. In *International Conference on Machine Learning*, pp. 2160–2169, 2019.

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.

Han Guo, Bowen Tan, Zhengzhong Liu, Eric Xing, and Zhiting Hu. Efficient (soft) Q-learning for text generation with limited good data. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 6969–6991, 2022.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *International Conference on Machine Learning*, pp. 1352–1361, 2017.

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.

Hamish Ivison, Junjie Oscar Yin, Rulin Shao, Teng Xiao, Nathan Lambert, and Hannaneh Hajishirzi. Tmax: A simple recipe for terminal agents. *arXiv preprint arXiv:2606.23321*, 2026.

Sham M. Kakade. A natural policy gradient. In *Advances in Neural Information Processing Systems*, volume 14, pp. 1531–1538, 2001.

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

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. VinePPO: Refining credit assignment in RL training of LLMs. In *International Conference on Machine Learning*, pp. 29557–29590, 2025.

Devvrit Khatri, Lovish Madaan, Rishabh Tiwari, Rachit Bansal, Sai Surya Duvvuri, Manzil Zaheer, Inderjit S. Dhillon, David Brandfonbrener, and Rishabh Agarwal. The art of scaling reinforcement learning compute for LLMs. In *International Conference on Learning Representations*, 2026.

Kimi Team. Kimi k1.5: Scaling reinforcement learning with LLMs. *arXiv preprint arXiv:2501.12599*, 2025.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tülu 3: Pushing frontiers in open language model post-training. In *Conference on Language Modeling*, 2025.

Guanghui Lan. Policy mirror descent for reinforcement learning: Linear convergence, new sampling complexity, and generalized problem classes. *Mathematical Programming*, 198(1):1059–1106, 2023.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q. Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. NuminaMath: The largest public dataset in AI4Maths with 860k pairs of competition math problems and solutions. Technical report, Project Numina, 2024.

Yu Li, Rui Miao, Tian Lan, and Zhengling Qi. OPPO: Bayesian value recursion for token-level credit assignment in LLM reasoning. *arXiv preprint arXiv:2605.21851*, 2026.

Ling Team, Anqi Shen, Baihui Li, Bin Hu, Bin Jing, Cai Chen, Chao Huang, Chao Zhang, Chaokun Yang, Cheng Lin, et al. Every step evolves: Scaling reinforcement learning for trillion-scale thinking model. *arXiv preprint arXiv:2510.18855*, 2025.

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. ProRL: Prolonged reinforcement learning expands reasoning boundaries in large language models. In *Advances in Neural Information Processing Systems*, volume 38, pp. 17998–18031, 2025a.

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. In *Conference on Language Modeling*, 2025b.

Chiyu Ma, Shuo Yang, Kexin Huang, Jinda Lu, Haoming Meng, Shangshang Wang, Bolin Ding, Soroush Vosoughi, Guoyin Wang, and Jingren Zhou. FIPO: Eliciting deep reasoning with future-KL influenced policy optimization. *arXiv preprint arXiv:2603.19835*, 2026a.

Wenhan Ma, Hailin Zhang, Liang Zhao, Yifan Song, Yudong Wang, Fuli Luo, and Zhifang Sui. Stabilizing MoE reinforcement learning by aligning training and inference routers. In *International Conference on Machine Learning*, 2026b.

Renjie Mao, Xiangxin Zhou, Lvfang Tao, Yixin Ding, Yu Shi, Yongguang Lin, Yuheng Wu, Honglin Zhu, Qian Qiu, and Wenxi Zhu. Beyond uniform token-level trust region in LLM reinforcement learning. *arXiv preprint arXiv:2606.10968*, 2026.

MiniMax, Aili Chen, Aonian Li, Bangwei Gong, Binyang Jiang, Bo Fei, Bo Yang, Boji Shan, Changqing Yu, Chao Wang, Cheng Zhu, et al. MiniMax-M1: Scaling test-time compute efficiently with lightning attention. *arXiv preprint arXiv:2506.13585*, 2025.

Ofir Nachum, Mohammad Norouzi, Kelvin Xu, and Dale Schuurmans. Bridging the gap between value and policy based reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Michael Noukhovitch, Shengyi Huang, Sophie Xhonneux, Arian Hosseini, Rishabh Agarwal, and Aaron Courville. Asynchronous RLHF: Faster and more efficient off-policy RL for language models. In *International Conference on Learning Representations*, 2025.

NVIDIA, Aaron Blakeman, Aaron Grattafiori, Aarti Basant, Abhibha Gupta, Abhinav Khattar, Adi Renduchintala, Aditya Vavre, Akanksha Shukla, Akhiad Bercovich, et al. NVIDIA Nemotron 3: Efficient and open intelligence. *arXiv preprint arXiv:2512.20856*, 2025.

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. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744, 2022.

Penghui Qi, Zichen Liu, Xiangxin Zhou, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Defeating the training-inference mismatch via FP16. *arXiv preprint arXiv:2510.26788*, 2025.

Penghui Qi, Xiangxin Zhou, Zichen Liu, Tianyu Pang, Chao Du, Min Lin, and Wee Sun Lee. Rethinking the trust region in LLM reinforcement learning. In *International Conference on Machine Learning*, 2026.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pp. 53728–53741, 2023.

Rafael Rafailov, Joey Hejna, Ryan Park, and Chelsea Finn. From r to Q^*: Your language model is secretly a Q-function. In *Conference on Language Modeling*, 2024.

Pierre Harvey Richemond, Yunhao Tang, Daniel Guo, Daniele Calandriello, Mohammad Gheshlaghi Azar, Rafael Rafailov, Bernardo Avila Pires, Eugene Tarassov, Lucas Spangher, Will Ellsworth, et al. Offline regularised reinforcement learning for large language models alignment. *arXiv preprint arXiv:2405.19107*, 2024.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pp. 1889–1897, 2015.

John Schulman, Xi Chen, and Pieter Abbeel. Equivalence between policy gradients and soft Q-learning. *arXiv preprint arXiv:1704.06440*, 2017a.

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

Lior Shani, Yonathan Efroni, and Shie Mannor. Adaptive trust region policy optimization: Global convergence and faster rates for regularized MDPs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5668–5675, 2020.

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

Charlie Snell, Ilya Kostrikov, Yi Su, Mengjiao Yang, and Sergey Levine. Offline RL for natural language generation with implicit language Q learning. In *International Conference on Learning Representations*, 2023.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2nd edition, 2018.

Manan Tomar, Lior Shani, Yonathan Efroni, and Mohammad Ghavamzadeh. Mirror descent policy optimization. In *International Conference on Learning Representations*, 2022.

Huaijie Wang, Shibo Hao, Hanze Dong, Shenao Zhang, Yilin Bao, Ziran Yang, and Yi Wu. Offline reinforcement learning for LLM multi-step reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 8881–8893, 2025.

Xumeng Wen, Zihan Liu, Shun Zheng, Shengyu Ye, Zhirong Wu, Yang Wang, Zhijian Xu, Xiao Liang, Junjie Li, Ziming Miao, et al. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base LLMs. In *International Conference on Learning Representations*, 2026.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256, 1992.

Yue Wu, Zhiqing Sun, Huizhuo Yuan, Kaixuan Ji, Yiming Yang, and Quanquan Gu. Self-play preference optimization for language model alignment. In *International Conference on Learning Representations*, 2025.

Lin Xiao. On the convergence rates of policy gradient methods. *Journal of Machine Learning Research*, 23(282):1–36, 2022.

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.

Feng Yao, Liyuan Liu, Dinghuai Zhang, Chengyu Dong, Jingbo Shang, and Jianfeng Gao. On the rollout-training mismatch in modern RL systems. In *OPT 2025: Optimization for Machine Learning (NeurIPS 2025 Workshop)*, 2025.

Jiarui Yao, Xiangxin Zhou, Penghui Qi, Wee Sun Lee, Liefeng Bo, and Tianyu Pang. Rethinking the divergence regularization in LLM RL. *arXiv preprint arXiv:2606.09821*, 2026.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, et al. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Yurun Yuan, Fan Chen, Zeyu Jia, Alexander Rakhlin, and Tengyang Xie. Trajectory Bellman residual minimization: A simple value-based method for LLM reasoning. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Yongcheng Zeng, Guoqing Liu, Weiyu Ma, Ning Yang, Haifeng Zhang, and Jun Wang. Token-level direct preference optimization. In *International Conference on Machine Learning*, pp. 58348–58365, 2024.

Wenhao Zhan, Shicong Cen, Baihe Huang, Yuxin Chen, Jason D. Lee, and Yuejie Chi. Policy mirror descent for regularized reinforcement learning: A generalized framework with linear convergence. *SIAM Journal on Optimization*, 33(2):1061–1091, 2023.

Yifan Zhang, Ge Zhang, Yue Wu, Kangping Xu, and Quanquan Gu. Beyond Bradley-Terry models: A general preference model for language model alignment. In *International Conference on Machine Learning*, pp. 76939–76965, 2025.

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. In *International Conference on Learning Representations*, 2026.

Chujie Zheng, Kai Dang, Bowen Yu, Mingze Li, Huiqiang Jiang, Junrong Lin, Yuqiong Liu, Hao Lin, Chencan Wu, Feng Hu, et al. 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, Jingren Zhou, and Junyang Lin. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025b.

Xiangxin Zhou, Jiarui Yao, Penghui Qi, Bowen Ping, Jiaqi Tang, Haonan Wang, and Tianyu Pang. Predictive divergence masks for LLM RL. *arXiv preprint arXiv:2607.10848*, 2026.

A Extended Related Work

Reinforcement learning with verifiable rewards. RLVR has become an important approach for improving the reasoning capabilities of LLMs (Lambert et al., 2025; Shao et al., 2024; Guo et al., 2025; Yang et al., 2025; Wen et al., 2026). Earlier RL fine-tuning of LLMs followed RLHF, which optimizes a learned reward model with PPO and a learned value network (Ouyang et al., 2022; Schulman et al., 2017b). On reasoning tasks, such value networks often give poor estimates of the expected return (Kazemnejad et al., 2025). Critic-free methods, which build on REINFORCE (Williams, 1992), have become the dominant choice in RLVR (Ahmadian et al., 2024; Shao et al., 2024; Liu et al., 2025b; Hu et al., 2025; Yu et al., 2025; Zheng et al., 2025b; MiniMax et al., 2025; Zhang et al., 2026), and GRPO has been scaled to large reasoning models (Guo et al., 2025; Yang et al., 2025). BPO is also critic-free and derives its per-token weight from PMD.

Policy mirror descent. PMD updates the policy at every state by maximizing the action value minus a Bregman divergence to the current policy (Geist et al., 2019; Tomar et al., 2022; Lan, 2023). With the KL divergence and a softmax policy, this update coincides with the natural policy gradient (Kakade, 2001; Agarwal et al., 2021), and Shani et al. (2020) analyze an adaptive variant of TRPO (Schulman et al., 2015) as PMD. Linear convergence of PMD and natural policy gradient has been established for both unregularized and regularized MDPs (Cen et al., 2022; Xiao, 2022; Lan, 2023; Zhan et al., 2023). For LLMs, several methods implement a mirror-descent step by regressing policy log-ratios onto rewards (Gao et al., 2024; Wu et al., 2025; Zhang et al., 2025; Kimi Team, 2025; Richemond et al., 2024; Brantley et al., 2025). REBEL regresses the difference in log-ratios between two responses onto their reward difference (Gao et al., 2024), and Kimi k1.5 optimizes a squared loss derived from online mirror descent (Kimi Team, 2025). DRO and A*-PO regress the log-ratio onto the reward minus a value estimate (Richemond et al., 2024; Brantley et al., 2025). All of these methods place the KL penalty on complete responses, so their target reweights the response distribution of an anchor policy by an exponential of the reward. BPO starts instead from the state-wise update (19), which reweights each next-token distribution of μ by the advantage A^{μ}.

Training-inference mismatch and policy lag. The rollout policy can differ from the policy being trained, because the inference and training engines compute token probabilities differently (Qi et al., 2025; Ma et al., 2026b) and because rollouts may come from earlier policy versions (Zheng et al., 2025a; Noukhovitch et al., 2025; Fu et al., 2025).

Bellman equations and implicit values. In entropy- and KL-regularized RL, the soft Bellman equation relates policy log-probabilities to value functions (Haarnoja et al., 2017; Schulman et al., 2017a; Nachum et al., 2017). Many methods for language models build on the Bellman equations, either to learn token-level values or to express rewards and values through policy log-ratios (Guo et al., 2022; Snell et al., 2023; Rafailov et al., 2023; 2024; Zeng et al., 2024; Cui et al., 2025; Wang et al., 2025; Clavier et al., 2025; Yuan et al., 2025; Kang et al., 2026; Li et al., 2026). BPO applies the Bellman equations of the rollout policy μ, and the minimizer of its objective is the PMD update from μ.

Trust regions and importance ratios. TRPO restricts each policy update to a KL trust region (Schulman et al., 2015), and PPO approximates this constraint by clipping the importance ratio of the sampled action (Schulman et al., 2017b). GRPO inherits token-level ratio clipping, and its variants change the clipping rule. DAPO decouples the lower and upper clipping ranges (Yu et al., 2025), GSPO clips a sequence-level ratio (Zheng et al., 2025b), and CISPO clips the importance-sampling weight instead of the token update (MiniMax et al., 2025). Qi et al. (2026) argue that the ratio of a single sampled token is a noisy estimate of the policy divergence under large vocabularies. They propose DPPO, which masks updates using a divergence estimated through binary or top-K approximations. Follow-up work predicts the direction in which the divergence will change (Zhou et al., 2026), adapts the threshold to token position and prefix drift (Mao et al., 2026), and replaces the hard mask with a divergence regularizer (Yao et al., 2026). BPO keeps the form of the GRPO mask but applies it to ω_t^i instead of the importance ratio. The weight ω_t^i follows from applying the binary KL approximation of Qi et al. (2026) to the critic-free PMD objective.

B Binary and Top-k KL Divergences

Binary KL divergence. Let $\pi, \mu \in \Pi$ be two policies on $\mathcal{V}$. For a fixed state s, their KL divergence is $D_{\mathrm{KL}}(\pi(\cdot|s) \,\|\, \mu(\cdot|s)) = \sum_{y\in\mathcal{V}} \pi(y|s) \log \frac{\pi(y|s)}{\mu(y|s)}$. Computing the full KL divergence requires logits for the entire vocabulary and can be costly.

The binary KL divergence introduced in Qi et al. (2026) approximates the full KL divergence. For a given action $y \in \mathcal{V}$, it partitions the action space into $\{y\}$ and its complement $\mathcal{V} \setminus \{y\}$. This induces the following Bernoulli distributions: $\pi_y^{\mathrm{bin}}(\cdot|s) = (\pi(y|s), 1-\pi(y|s))$, $\mu_y^{\mathrm{bin}}(\cdot|s) = (\mu(y|s), 1-\mu(y|s))$. We define the binary KL divergence associated with action y as

$$\begin{aligned} D_{\mathrm{KL}}^{\mathrm{bin}}\left(\pi(\cdot|s) \,\|\, \mu(\cdot|s); y\right) &= D_{\mathrm{KL}}\left(\pi_y^{\mathrm{bin}}(\cdot|s) \,\|\, \mu_y^{\mathrm{bin}}(\cdot|s)\right) \\ &= \pi(y|s) \log \frac{\pi(y|s)}{\mu(y|s)} + (1-\pi(y|s)) \log \frac{1-\pi(y|s)}{1-\mu(y|s)}. \end{aligned} \tag{17}$$

As shown in Qi et al. (2026), binary KL divergence provides a lower bound for KL divergence, i.e.,

$$0 \le D_{\mathrm{KL}}^{\mathrm{bin}}\left(\pi(\cdot|s) \,\|\, \mu(\cdot|s); y\right) \le D_{\mathrm{KL}}\left(\pi(\cdot|s) \,\|\, \mu(\cdot|s)\right), \quad \forall y \in \mathcal{V}.$$

Top-k KL divergence. Following Qi et al. (2026), the top-k approximation tracks the k most probable tokens under the rollout policy. For a state s and a sampled action $y \in \mathcal{V}$, let

$$\mathcal{V}_k(s; y) = \mathrm{TopK}\left(\mu(\cdot|s), k\right) \cup \{y\},$$

which contains the k highest-probability tokens under $\mu(\cdot|s)$, augmented with y if it is not already present. For $\tilde{\pi} \in \{\mu, \pi\}$, the reduced distribution $p_s^{\tilde{\pi}}$ over $\mathcal{V}_k(s; y) \cup \{\mathrm{other}\}$ keeps the original probability of each token in $\mathcal{V}_k(s; y)$ and aggregates all other tokens into the category "other":

$$p_s^{\tilde{\pi}}(a) = \tilde{\pi}(a|s), \quad \forall a \in \mathcal{V}_k(s; y), \qquad p_s^{\tilde{\pi}}(\mathrm{other}) = 1 - \sum_{a\in\mathcal{V}_k(s;y)} \tilde{\pi}(a|s).$$

The top-k KL divergence is the KL divergence between the reduced distributions:

$$D_{\mathrm{KL}}^{\text{top-}k}\left(\mu(\cdot|s) \,\|\, \pi(\cdot|s); y\right) = \sum_{a\in\mathcal{V}_k(s;y)\cup\{\mathrm{other}\}} p_s^{\mu}(a) \log \frac{p_s^{\mu}(a)}{p_s^{\pi}(a)}. \tag{18}$$

Like the binary approximation, the top-k approximation corresponds to a partition of the vocabulary and is a lower bound of the full KL divergence (Qi et al., 2026).

C Omitted Derivations and Proofs

C.1 Advantage-Based Policy Mirror Descent

Advantage-based PMD solves the following problem for each state $s_t = (x, y_{<t})$:

$$\max_{\pi(\cdot|s_t)\in\Delta(\mathcal{V})} \quad \mathbb{E}_{y_t\sim\pi(\cdot|s_t)}\left[A^\mu(s_t, y_t)\right] - \frac{1}{\eta} D_{\mathrm{KL}}(\pi(\cdot|s_t) \,\|\, \mu(\cdot|s_t)). \tag{19}$$

As discussed in Section 2.4, the optimization problem (19) has the unique solution π^+, given at each state $s_t = (x, y_{<t})$ by

$$\pi^+(y_t|s_t) = \frac{\mu(y_t|s_t)\exp\left(\eta A^\mu(s_t, y_t)\right)}{Z_\mu(s_t)}, \tag{20}$$

where the partition function is $Z_\mu(s_t) = \mathbb{E}_{y_t'\sim\mu(\cdot|s_t)}\exp\left(\eta A^\mu(s_t, y_t')\right)$. Here, we use π^+ and Z_μ as abbreviations for π^+_{μ,A^μ} and Z_{μ,A^μ}.

C.2 Derivation of the Critic-Free Objective (11) from Advantage-Based Policy Mirror Descent

We derive the critic-free objective (11) from the original advantage-based PMD (19) in three steps.

• Step 1: We rewrite (20) as the PMD optimality condition and eliminate the state-dependent partition function $Z_\mu(s_t)$. For any prompt x, response y, and step t with $1 \le t \le |y|$, we can rewrite (20) as

$$\log \pi^+(y_t|s_t) - \log \mu(y_t|s_t) - \eta A^\mu(s_t, y_t) + \log Z_\mu(s_t) = 0. \tag{21}$$

Taking the expectation over $y_t \sim \mu(\cdot|s_t)$ and using (6) gives

$$\log Z_\mu(s_t) = D_{\mathrm{KL}}(\mu(\cdot|s_t) \,\|\, \pi^+(\cdot|s_t)) + \eta \sum_{y_t\in\mathcal{V}} \mu(y_t|s_t) A^\mu(s_t, y_t) = D_{\mathrm{KL}}(\mu(\cdot|s_t) \,\|\, \pi^+(\cdot|s_t)). \tag{22}$$

Then, substituting (22) into (21) yields

$$\log \pi^+(y_t|s_t) - \log \mu(y_t|s_t) - \eta A^\mu(s_t, y_t) + D_{\mathrm{KL}}(\mu(\cdot|s_t) \,\|\, \pi^+(\cdot|s_t)) = 0. \tag{23}$$

• Step 2: We next express the cumulative token-level advantages in terms of the terminal reward. By the Bellman equation, substituting (4) into (5) gives

$$A^\mu(s_t, y_t) = V^\mu(s_{t+1}) - V^\mu(s_t).$$

Summing over the trajectory gives

$$\sum_{t=1}^{|y|} A^\mu(s_t, y_t) = V^\mu(s_{|y|+1}) - V^\mu(s_1) = R(x, y) - V^\mu(x). \tag{24}$$

• Step 3: For any prompt-response pair (x, y), summing (23) over the trajectory and using (24) gives

$$\sum_{t=1}^{|y|} \left(\log \pi^+(y_t|s_t) - \log \mu(y_t|s_t) + D_{\mathrm{KL}}(\mu(\cdot|s_t) \,\|\, \pi^+(\cdot|s_t))\right) = \eta\left(R(x, y) - V^\mu(x)\right),$$

i.e.,

$$\delta(x, y; \pi^+, \mu) = 0. \tag{25}$$

Minimizing the expected squared residual under the rollout distribution gives (11).

C.3 Proof of Theorem 1

Proof of Sufficiency. We first show that the PMD solution π^+ minimizes the reformulated objective (11). By (25), $\delta(x, y; \pi^+, \mu) = 0$ for every prompt-response pair (x, y), so π^+ attains objective value zero. Since the objective is nonnegative, π^+ is an optimal solution of (11).

Proof Sketch of Necessity. It remains to show that every optimal solution of (11) coincides with π^+ on all states reachable under μ. Define the token-level residual

$$d_\pi(s_t, y_t) = \eta A^\mu(s_t, y_t) - \Big(\log \pi(y_t|s_t) - \log \mu(y_t|s_t) + D_{\mathrm{KL}}(\mu(\cdot|s_t) \,\|\, \pi(\cdot|s_t))\Big).$$

For any feasible policy π for which the above quantities are finite, its conditional expectation under the rollout policy is zero:

$$\begin{aligned}
&\mathbb{E}_{y_t \sim \mu(\cdot|s_t)}\left[d_\pi(s_t, y_t)\right] \\
&= \eta \sum_{y_t \in \mathcal{V}} \mu(y_t|s_t) A^\mu(s_t, y_t) - \sum_{y_t \in \mathcal{V}} \mu(y_t|s_t) \log \frac{\pi(y_t|s_t)}{\mu(y_t|s_t)} - D_{\mathrm{KL}}(\mu(\cdot|s_t) \,\|\, \pi(\cdot|s_t)) \\
&= 0.
\end{aligned}$$

Therefore, the cumulative token-level residuals form a martingale under the rollout policy μ. For any optimal solution of (11), the terminal value of this martingale equals the trajectory-level residual and is zero almost surely. Since the response length is bounded by $T_{\max}$, the martingale property then implies that every partial sum is zero almost surely. Thus, each token-level residual vanishes. We now give the complete proof.

Complete Proof of Necessity. We use $\mathbb{I}_A$ to denote the indicator of an event A throughout this section.

Let $\mathrm{supp}\mu(\cdot|s)$ denote the set of actions with positive probability under policy μ at state s. Let $\mathrm{supp}\mathbb{P}_\mu(\cdot|x)$ denote the set of responses with positive probability under policy μ given prompt x. Let $\mathcal{S}_x^\mu$ denote the set of states reachable from prompt x with positive probability under policy μ.

Fix any prompt $x \in D$. For any state-action pair (s, a), define

$$d_\pi(s, a) = \eta A^\mu(s, a) - \Big(\log \pi(a|s) - \log \mu(a|s) + D_{\mathrm{KL}}(\mu(\cdot|s) \,\|\, \pi(\cdot|s))\Big).$$

By definition,

$$\delta(x, y; \pi, \mu) = \sum_{t=1}^{|y|} d_\pi(s_t, y_t). \tag{26}$$

We first show that the minimum of (11) is zero. The PMD optimality condition (23) gives

$$d_{\pi^+}(s, a) = 0.$$

The exponential-form update (20) preserves the support of the rollout policy, so $\pi^+ \in \Pi_\mu$ and

$$\delta(x, y; \pi^+, \mu) = 0, \quad \forall y \in \mathrm{supp}\mathbb{P}_\mu(\cdot|x).$$

Hence, π^+ is feasible for (11) and achieves objective value zero. Since the objective is nonnegative, the minimum value of (11) is zero.

Now let $\hat{\pi}^+ \in \Pi_\mu$ be any optimal solution of (11). Since the minimum value is zero and $\phi(x) > 0$, we have

$$\delta(x, Y; \hat{\pi}^+, \mu) = 0, \quad Y \sim \mathbb{P}_\mu(\cdot|x)\text{-almost surely.} \tag{27}$$

Since the objective value is finite, for every state $s \in \mathcal{S}_x^\mu$, $D_{\mathrm{KL}}(\mu(\cdot|s) \,\|\, \hat{\pi}^+(\cdot|s)) < \infty$. Consequently, $\mu(\cdot|s)$ is absolutely continuous with respect to $\hat{\pi}^+(\cdot|s)$ for any $s \in \mathcal{S}_x^\mu$. Thus, all quantities below are finite on states reachable under μ.

Let τ be the time at which generation terminates:

$$\tau = \inf\{t \geq 1 : Y_t = \mathrm{EOS}\} \wedge T_{\max}.$$

Define the filtration

$$\mathcal{F}_t = \sigma\left(Y_1, \ldots, Y_{t\wedge\tau}\right), \quad 0 \le t \le T_{\max}.$$

For $1 \le t \le T_{\max}$, define

$$D_t = \mathbb{I}_{\{t\le\tau\}} d_{\hat{\pi}^+}(S_t, Y_t), \quad M_n = \sum_{t=1}^{n} D_t,$$

where $S_t = (x, Y_{<t})$ on the event $\{t \le \tau\}$.

Since $\mathbb{I}_{\{t\le\tau\}}$ and S_t are $\mathcal{F}_{t-1}$-measurable, we have

$$\begin{aligned}
&\mathbb{E}\left[D_t \mid \mathcal{F}_{t-1}\right] \\
&= \mathbb{I}_{\{t\le\tau\}} \sum_{a\in\mathcal{V}} \mu(a|S_t) d_{\hat{\pi}^+}(S_t, a) \\
&= \mathbb{I}_{\{t\le\tau\}} \eta \sum_{a\in\mathcal{V}} \mu(a|S_t) A^{\mu}(S_t, a) \\
&\quad - \mathbb{I}_{\{t\le\tau\}} \Big(\sum_{a\in\mathcal{V}} \mu(a|S_t) \log \frac{\hat{\pi}^+(a|S_t)}{\mu(a|S_t)} + D_{\mathrm{KL}}(\mu(\cdot|S_t) \,\|\, \hat{\pi}^+(\cdot|S_t)) \Big) \\
&= 0,
\end{aligned}$$

where the last equality follows from (6). Thus, $\{M_n\}_{n=0}^{T_{\max}}$ is an integrable martingale.

By (26) and (27),

$$M_{T_{\max}} = \sum_{t=1}^{\tau} d_{\hat{\pi}^+}(S_t, Y_t) = \delta(x, Y; \hat{\pi}^+, \mu) = 0, \quad \mathbb{P}_{\mu}(\cdot|x)\text{-almost surely.}$$

Thus, for any $0 \le n \le T_{\max}$,

$$M_n = \mathbb{E}\left[M_{T_{\max}} | \mathcal{F}_n\right] = 0, \quad \mathbb{P}_{\mu}(\cdot|x)\text{-almost surely.}$$

It follows that for any $1 \le t \le T_{\max}$,

$$D_t = M_t - M_{t-1} = 0 \quad \mathbb{P}_{\mu}(\cdot|x)\text{-almost surely.}$$

Consider any state $s \in \mathcal{S}_x^{\mu}$ and any action $a \in \mathrm{supp}\mu(\cdot|s)$. For the corresponding step t, the event $\{S_t = s, Y_t = a\}$ has strictly positive probability under $\mathbb{P}_{\mu}(\cdot|x)$. Since $D_t = 0$ almost surely, this implies

$$d_{\hat{\pi}^+}(s, a) = 0.$$

Combining this with $d_{\pi^+}(s, a) = 0$, we obtain

$$\log \frac{\hat{\pi}^+(a|s)}{\pi^+(a|s)} = D_{\mathrm{KL}}(\mu(\cdot|s) \,\|\, \pi^+(\cdot|s)) - D_{\mathrm{KL}}(\mu(\cdot|s) \,\|\, \hat{\pi}^+(\cdot|s)).$$

The right-hand side does not depend on a. Hence, for every $s \in \mathcal{S}_x^{\mu}$, there exists a constant $c(s) > 0$ such that

$$\hat{\pi}^+(a|s) = c(s)\pi^+(a|s), \quad \forall a \in \mathrm{supp}\mu(\cdot|s).$$

Since $\hat{\pi}^+, \pi^+ \in \Pi_{\mu}$, both policies assign zero probability outside $\mathrm{supp}\mu(\cdot|s)$. Therefore,

$$1 = \sum_{a\in\mathrm{supp}\mu(\cdot|s)} \hat{\pi}^+(a|s) = c(s) \sum_{a\in\mathrm{supp}\mu(\cdot|s)} \pi^+(a|s) = c(s).$$

Thus, $c(s) = 1$, and

$$\hat{\pi}^+(\cdot|s) = \pi^+(\cdot|s), \quad \forall s \in \mathcal{S}_x^{\mu}.$$

In addition, since $\hat{\pi}^+, \pi^+ \in \Pi_{\mu}$, both π^+ and $\hat{\pi}^+$ assign zero probability outside $\mathrm{supp}\mathbb{P}_{\mu}(\cdot|x)$. Consequently,

$$\mathbb{P}_{\hat{\pi}^+}(\cdot|x) = \mathbb{P}_{\pi^+}(\cdot|x).$$

This completes the proof. □

C.4 Proof of the Identity (15)

Fix a state s and an action y, and let

$$p = \pi(y|s), \quad q = \mu(y|s).$$

The binary KL divergence associated with action y is

$$D_{\mathrm{KL}}^{\mathrm{bin}}\left(\mu(\cdot|s) \,\|\, \pi(\cdot|s); y\right) = q \log \frac{q}{p} + (1-q) \log \frac{1-q}{1-p}.$$

Differentiating with respect to π gives

$$\nabla D_{\mathrm{KL}}^{\mathrm{bin}}\left(\mu(\cdot|s) \,\|\, \pi(\cdot|s); y\right) = -q \nabla \log p + \frac{1-q}{1-p} \nabla p.$$

Using the fact that $\nabla p = p \nabla \log p$, we obtain

$$\begin{aligned} &\nabla \Big(\log p + D_{\mathrm{KL}}^{\mathrm{bin}}\left(\mu(\cdot|s) \,\|\, \pi(\cdot|s); y\right) \Big) \\ &= (1-q) \nabla \log p + \frac{(1-q)p}{1-p} \nabla \log p = \frac{1-\mu(y|s)}{1-\pi(y|s)} \nabla \log \pi(y|s), \end{aligned}$$

i.e.,

$$\nabla \Big(\log \pi(y|s) + D_{\mathrm{KL}}^{\mathrm{bin}}\left(\mu(\cdot|s) \,\|\, \pi(\cdot|s); y\right) \Big) = \frac{1-\mu(y|s)}{1-\pi(y|s)} \nabla \log \pi(y|s).$$

This proves (15). □

D Details of the Compared IS-Based Methods

The complete formulations of GRPO, CISPO, and DPPO are given in Appendix C.1 of Qi et al. (2026). We include the mask M_t^i and cap C of these methods in the unified form (1), together with the sequence-level ratio of GSPO, here for completeness.

GRPO. GRPO (Shao et al., 2024) is a critic-free, PPO-style policy optimization method (Schulman et al., 2017b). GRPO applies PPO-style clipping to the token-level IS ratio r_t^i in (2). For the t-th token in response y^i, its per-token loss is defined as

$$\mathcal{L}_{i,t}^{\mathrm{GRPO}}(\pi) = -\min \left\{ r_t^i \hat{A}^i,\ \mathrm{clip}\left(r_t^i, 1-\epsilon_{\mathrm{low}}, 1+\epsilon_{\mathrm{high}}\right) \hat{A}^i \right\}. \tag{28}$$

The following per-token loss has the same gradient as (28) and is therefore equivalent for gradient-based training:

$$\widetilde{\mathcal{L}}_{i,t}^{\mathrm{GRPO}}(\pi) = -\hat{A}^i M_t^i \,\mathrm{sg}\left(r_t^i\right) \log \pi(y_t^i|s_t^i), \tag{29}$$

where the clipping mask M_t^i is defined as

$$M_t^i = \begin{cases} 0, & \hat{A}^i > 0 \text{ and } r_t^i > 1+\epsilon_{\mathrm{high}}, \\ 0, & \hat{A}^i < 0 \text{ and } r_t^i < 1-\epsilon_{\mathrm{low}}, \\ 1, & \text{otherwise.} \end{cases} \tag{30}$$

Thus, GRPO corresponds to (1) with the mask M_t^i in (30) and $C = \infty$. GRPO-ClipHigher uses the asymmetric clipping interval $[1-\epsilon_{\mathrm{low}}, 1+\epsilon_{\mathrm{high}}] = [0.8, 1.28]$, i.e., $\epsilon_{\mathrm{low}} = 0.2$ and $\epsilon_{\mathrm{high}} = 0.28$ (Yu et al., 2025).

CISPO. CISPO (MiniMax et al., 2025) corresponds to (1) with $M_t^i = 1$ and a finite cap C.

DPPO. DPPO (Qi et al., 2026) corresponds to (1) with $C = \infty$ and a mask conditioned on the policy divergence D_t^i between $\mu(\cdot|s_t^i)$ and $\pi(\cdot|s_t^i)$:

$$M_t^i = \begin{cases} 0, & \hat{A}^i > 0,\ r_t^i > 1, \text{ and } D_t^i > \delta, \\ 0, & \hat{A}^i < 0,\ r_t^i < 1, \text{ and } D_t^i > \delta, \\ 1, & \text{otherwise,} \end{cases}$$

where $\delta > 0$ is the divergence threshold. We use the binary total variation approximation of Qi et al. (2026), i.e., $D_t^i = \left|\mu(y_t^i|s_t^i) - \pi(y_t^i|s_t^i)\right|$.

GSPO. GSPO (Zheng et al., 2025b) extends importance sampling to the sequence level. It replaces the token-level ratio r_t^i with the length-normalized sequence-level ratio

$$s^i = \left(\frac{\mathbb{P}_\pi(y^i|x)}{\mathbb{P}_\mu(y^i|x)}\right)^{1/|y^i|} = \exp\left(\frac{1}{|y^i|}\sum_{t=1}^{|y^i|}\log r_t^i\right),$$

so that all tokens of y^i share the same ratio, and it applies clipping to s^i for the entire response.

E Clipping Mask of BPO

The mask M_t^i in the BPO loss (9) follows the GRPO clipping rule in (30), with the weight ω_t^i in place of the importance-sampling ratio r_t^i:

$$M_t^i = \begin{cases} 0, & \hat{A}^i > 0 \text{ and } \omega_t^i > 1 + \epsilon_{\text{high}} \\ 0, & \hat{A}^i < 0 \text{ and } \omega_t^i < 1 - \epsilon_{\text{low}} \\ 1, & \text{otherwise.} \end{cases} \tag{31}$$

F Experimental Details

Table 3 lists the hyperparameters that differ across experimental settings. These hyperparameters are shared by all methods within a setting. Table 4 lists the remaining hyperparameters, which are shared by all experiments. All methods use `seq-mean-token-mean` loss aggregation. All training runs with mixture-of-experts (MoE) models use Rollout Routing Replay (R3) (Ma et al., 2026b). The tool-integrated setting uses the ReTool-SFT dataset for cold-start fine-tuning.[1] In this setting, each response contains at most 16 generation turns, and each turn executes at most one Python tool call. Let $R_i = R(x, y^i)$ denote the reward of response y^i. Following GRPO (Shao et al., 2024), all methods assign each response the group-normalized advantage

$$\hat{A}^i = \frac{R_i - \texttt{mean}\big(\{R_j\}_{j=1}^G\big)}{\texttt{std}\big(\{R_j\}_{j=1}^G\big)}. \tag{32}$$

The AIME benchmarks are available on Hugging Face.[2] We evaluate on AIME24–26 every ten rollout batches (i.e., training steps). We evaluate on AMC23, HMMT25, and HMMT26 at the reported checkpoints. Evaluation uses the same response-length limit as training for each model.

We report Avg@32 as an estimate of Pass@1. In Tables 1, 2, 5, 6, and 9, the Avg. column is the arithmetic mean of the Avg@32 scores on the six benchmarks. In all training curves, the average accuracy is the arithmetic mean of the AIME24, AIME25, and AIME26 Avg@32 scores.

For results at the best checkpoint, we select the checkpoint with the highest mean Avg@32 across AIME24–26, choosing the earlier step in a tie.

GRPO-ClipHigher uses the asymmetric clipping interval $[0.8, 1.28]$ from DAPO (Yu et al., 2025). CISPO uses an importance-weight cap of 3.0, following the configuration used in the stability experiments of DPPO (Qi et al., 2026). DPPO uses binary total variation with threshold $\delta = 0.1$, following Tmax (Ivison et al., 2026). GSPO uses the clipping interval $[1 - 3 \times 10^{-3}, 1 + 5 \times 10^{-3}]$, following the ablation results in Khatri et al. (2026). BPO uses the default setting $\epsilon = 0.1$ and $C = 3.0$ in the main experiments. For its clipping mask M_t^i, BPO directly uses the interval $[0.8, 1.28]$ of GRPO-ClipHigher, i.e., $\epsilon_{\text{low}} = 0.2$ and $\epsilon_{\text{high}} = 0.28$. We made no attempt to tune this interval for BPO.

[1] `https://huggingface.co/datasets/JoeYing/ReTool-SFT`.

[2] AIME24: `https://huggingface.co/datasets/math-ai/aime24`, AIME25: `https://huggingface.co/datasets/math-ai/aime25`, AIME26: `https://huggingface.co/datasets/math-ai/aime26`.

Table 3: Hyperparameters that differ across experimental settings. All methods within a setting share these hyperparameters.

HYPERPARAMETER	LONG-COT	TOOL-INTEGRATED	ABLATION ON BPO HYPERPARAMETERS
Base model	Qwen3-30B-A3B-Base	Qwen3-4B-Instruct-2507	Qwen3-4B-Base
Max response length	16384	16384	8192
Prompts per batch	256	128	128
Responses per prompt	16	16	16
Minibatches per step	8	4	4
Responses per minibatch	512	512	512

Table 4: Hyperparameters shared by all experiments.

HYPERPARAMETER	VALUE
Training data	DAPO-Math-17k (English subset)
Optimizer	AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.98$)
Learning rate	10^{-6} (constant)
Weight decay	0.1
Gradient clipping	1.0
KL coefficient	0
Entropy coefficient	0
Temperature	1.0 (training and evaluation)
Top-p	1.0 (training and evaluation)
Evaluation interval	10 training steps
Evaluation responses per question	32

G Additional Results

G.1 Results at the Best Checkpoint

Tables 5 and 6 report results at the checkpoint with the highest mean accuracy over AIME24–26 for the two settings in Section 4. In the long-CoT setting, BPO achieves the highest average accuracy of 45.9%, 2.3 percentage points higher than CISPO. In the tool-integrated setting, BPO achieves the highest average accuracy of 65.2%, compared with 64.6% for CISPO.

Table 5: Avg@32 (%) on Qwen3-30B-A3B-Base at the best checkpoint of each method. The best result in each column is bold and the second best is underlined.

METHOD	AMC23	AIME24	AIME25	AIME26	HMMT25	HMMT26	AVG.
GRPO-ClipHigher	88.3	45.6	34.8	38.0	13.1	9.7	38.3
GSPO	<u>93.4</u>	50.3	35.5	44.6	**17.2**	7.8	41.5
CISPO	90.8	52.7	39.0	<u>50.4</u>	16.6	<u>12.2</u>	<u>43.6</u>
DPPO	92.3	<u>55.8</u>	<u>39.2</u>	44.2	13.0	10.6	42.5
BPO	**94.1**	**57.4**	**41.0**	**53.0**	<u>17.1</u>	**12.7**	**45.9**

Table 6: Avg@32 (%) of multi-turn tool-integrated reasoning on Qwen3-4B-Instruct-2507 at the best checkpoint of each method. The best result in each column is bold and the second best is underlined.

METHOD	AMC23	AIME24	AIME25	AIME26	HMMT25	HMMT26	AVG.
GRPO-ClipHigher	95.2	65.7	60.2	64.1	41.0	<u>44.4</u>	61.8
CISPO	**97.6**	**70.1**	<u>62.8</u>	67.3	**45.5**	<u>44.4</u>	<u>64.6</u>
DPPO	96.1	64.3	61.7	<u>68.4</u>	42.3	43.3	62.7
BPO	<u>96.9</u>	<u>69.5</u>	**64.2**	**71.8**	<u>43.5</u>	**45.2**	**65.2**

G.2 Mismatch-Correction Weights across Training Steps

Figure 4 shows the mismatch-correction weights of Section 4.2 at training steps 99, 199, 299, and 399, and Figure 5 shows the corresponding densities.

G.3 Training-Inference Mismatch

Figure 6 shows the mismatch between the current policy π and the rollout policy μ during the long-CoT training in Section 4.1. For BPO, DPPO, CISPO, and GRPO-ClipHigher, which operate at the token level, the y-axis is $|\log \pi(y_t^i|s_t^i) - \log \mu(y_t^i|s_t^i)|$ averaged over all tokens in the batch. Because GSPO operates at the sequence level, for GSPO we first average $\log \pi(y_t^i|s_t^i) - \log \mu(y_t^i|s_t^i)$ over the tokens of each response, then take the absolute value, and finally average over responses in the batch. Positive and negative token-level differences cancel within a response, so the GSPO curve is much lower and is not directly comparable to the other curves. Among the token-level methods, BPO has the smallest and most stable mismatch throughout training.

H Ablation Studies

We study the two BPO hyperparameters ϵ and C on Qwen3-4B-Base for 1000 training steps with the setting of the ablation on BPO hyperparameters in Table 3. Each sweep changes only one hyperparameter while keeping the other settings fixed. Both sweeps in Section H.1 and Section H.2 share the same GRPO-ClipHigher baseline. We also ablate three design choices of BPO on Qwen3-30B-A3B-Base with the long-CoT setting in Table 3 (Sections H.4–H.5). The figures in these sections show the first 200 training steps.

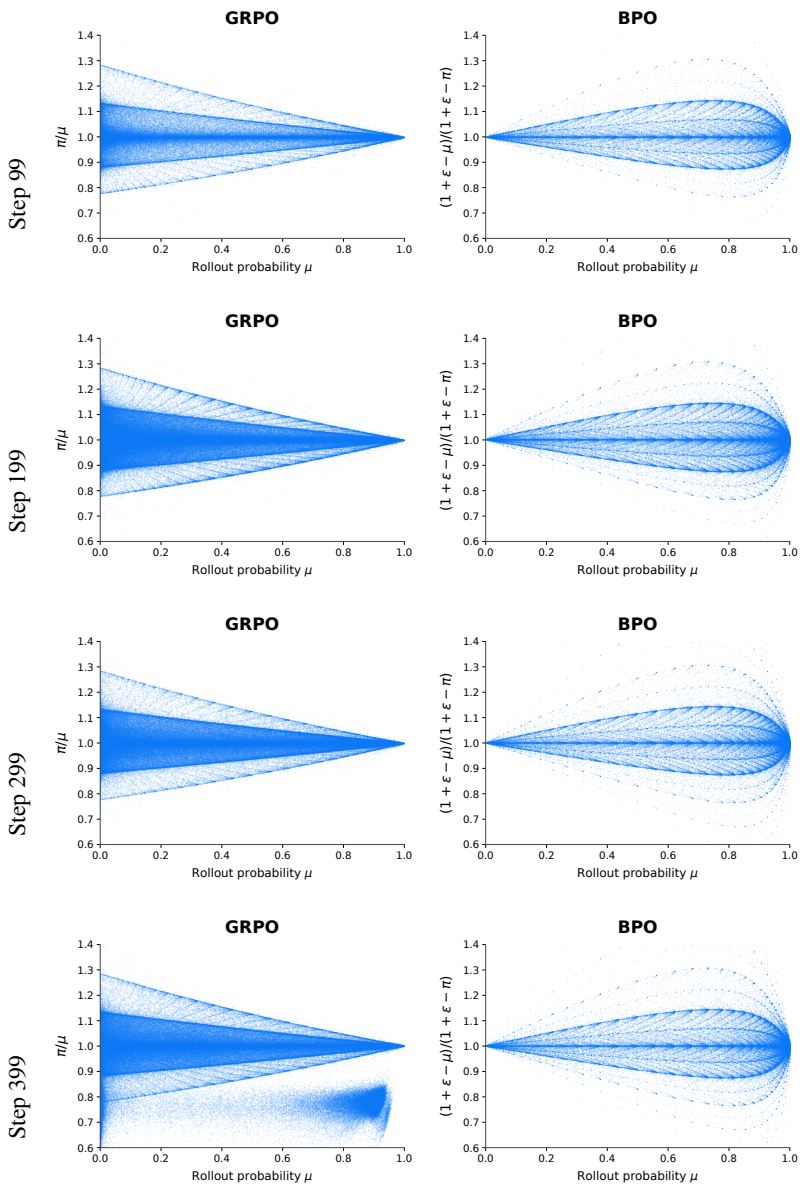

Figure 4: Mismatch-correction weight versus rollout probability μ on Qwen3-30B-A3B-Base. Left: importance-sampling ratio π/μ of GRPO-ClipHigher. Right: BPO weight $(1+\epsilon-\mu)/(1+\epsilon-\pi)$. Rows correspond to training steps 99, 199, 299, and 399. Each panel shows 10^5 tokens uniformly subsampled from the rollout batch of that step.

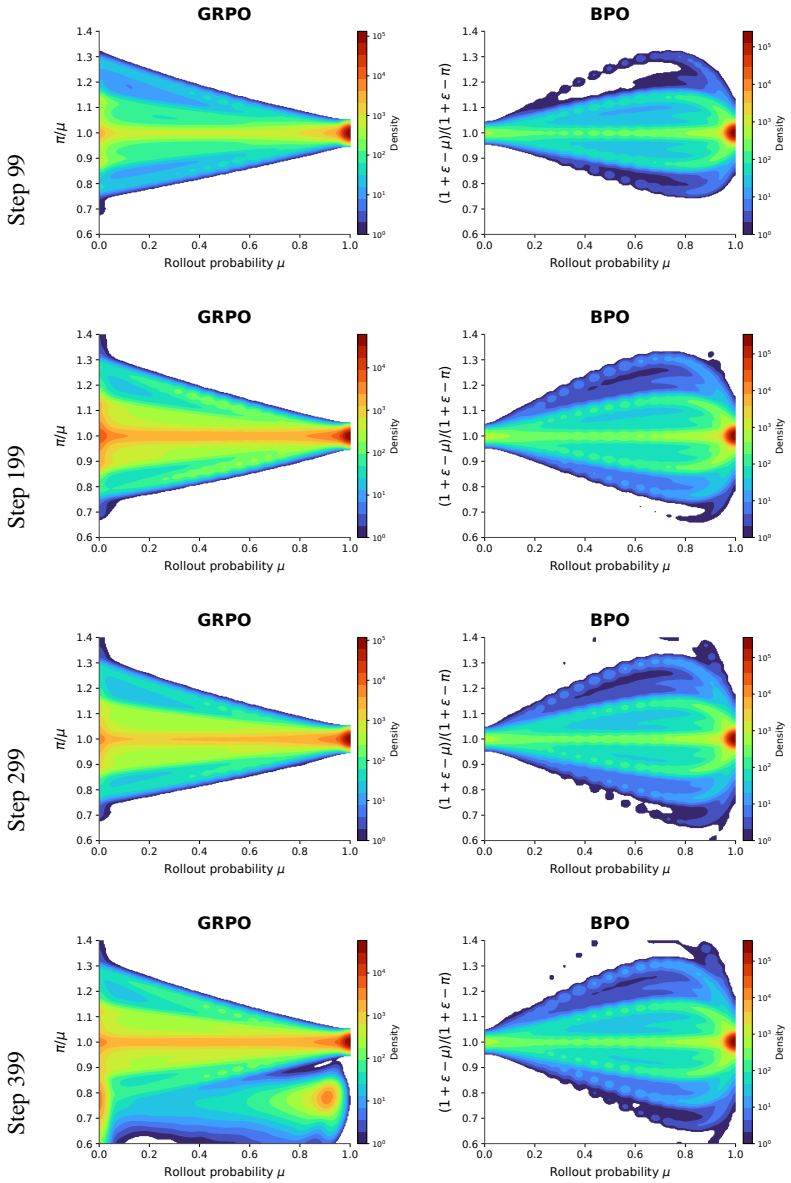

Figure 5: Density of the mismatch-correction weight versus rollout probability μ on Qwen3-30B-A3B-Base. Left: importance-sampling ratio π/μ of GRPO-ClipHigher. Right: BPO weight $(1+\epsilon-\mu)/(1+\epsilon-\pi)$. Rows correspond to training steps 99, 199, 299, and 399. Colors show density on a logarithmic scale.

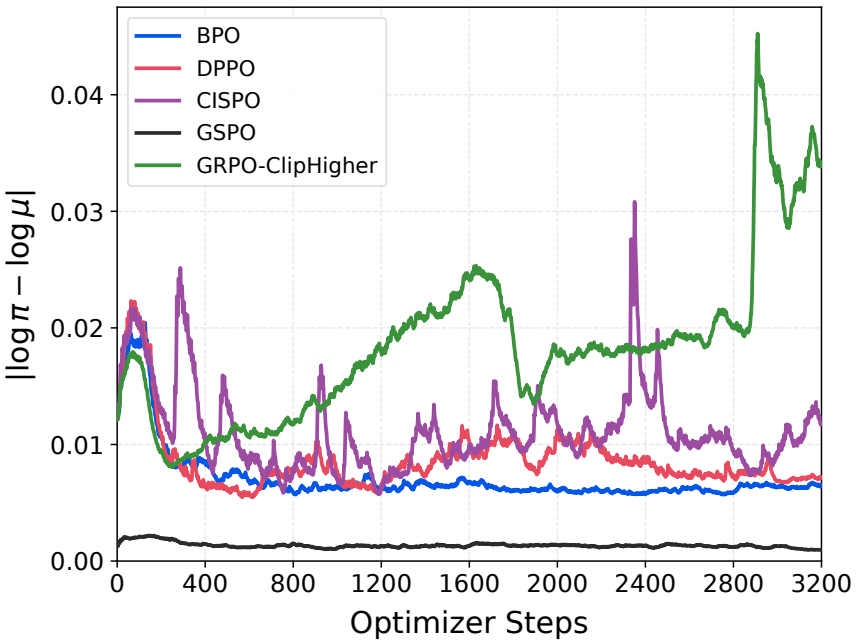

Figure 6: Training-inference mismatch on Qwen3-30B-A3B-Base. Curves are smoothed with an exponential moving average (EMA) with decay rate 0.95. The x-axis counts optimizer updates, with eight updates per training step.

Table 7: Ablation on BPO's smoothing parameter ϵ with $C = 3.0$. Each entry is the peak Avg@32 (%) on the benchmark within the first 1000 training steps. The best result in each column is bold and the second best is underlined.

METHOD	AIME24	AIME25	AIME26
BPO ($\epsilon = 0.05$)	**31.9**	23.9	<u>23.3</u>
BPO ($\epsilon = 0.1$)	28.4	**28.0**	<u>23.3</u>
BPO ($\epsilon = 0.2$)	<u>30.0</u>	26.4	**23.6**
BPO ($\epsilon = 0.3$)	27.5	<u>26.9</u>	20.7
GRPO-ClipHigher	22.1	23.4	16.6

H.1 Sensitivity to ϵ

We fix $C = 3.0$ and vary ϵ over $\{0.05, 0.1, 0.2, 0.3\}$. Figure 7 shows the training curves. Table 7 reports the peak Avg@32 of each setting within the first 1000 training steps, separately on AIME24, AIME25, and AIME26.

As shown in Figure 7, all four settings outperform GRPO-ClipHigher. For $\epsilon \in \{0.05, 0.1, 0.2\}$, the peak mean accuracy over AIME24–26 ranges from 25.4% to 25.8%, a difference of 0.4 percentage points, and is 4.9–5.3 points higher than that of GRPO-ClipHigher (20.5%). With $\epsilon = 0.3$, the peak is slightly lower at 24.1%, still 3.6 points higher than GRPO-ClipHigher. Table 7 shows that all four settings also outperform GRPO-ClipHigher on each of AIME24, AIME25, and AIME26.

H.2 Sensitivity to C

We fix $\epsilon = 0.1$ and vary C over $\{2.0, 3.0, 4.0\}$. Figure 8 shows the training curves. Table 8 reports the peak Avg@32 of each setting within the first 1000 training steps, separately on AIME24, AIME25, and AIME26.

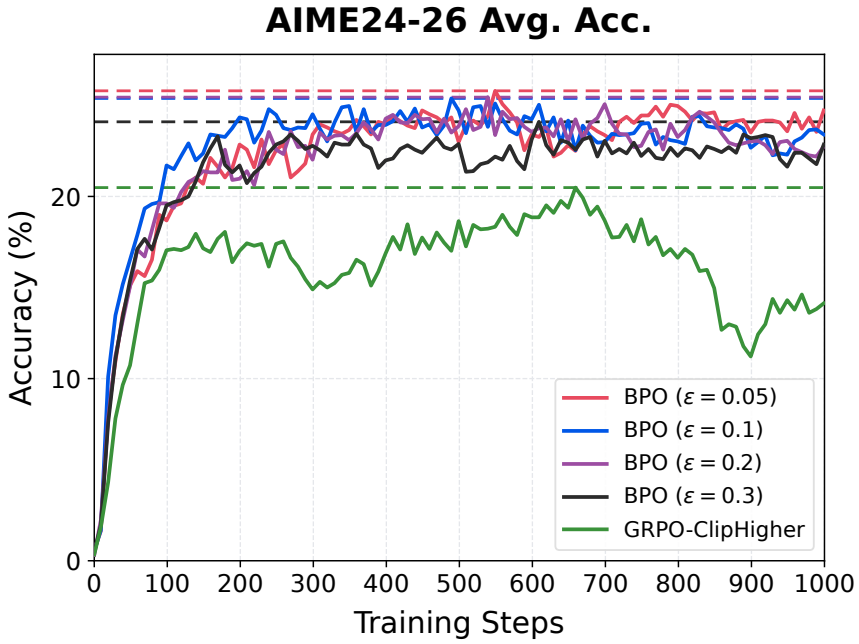

Figure 7: Effect of BPO's smoothing parameter ϵ on Qwen3-4B-Base with $C = 3.0$. Curves show mean Avg@32 across AIME24–26. Dashed lines mark the peak of each curve.

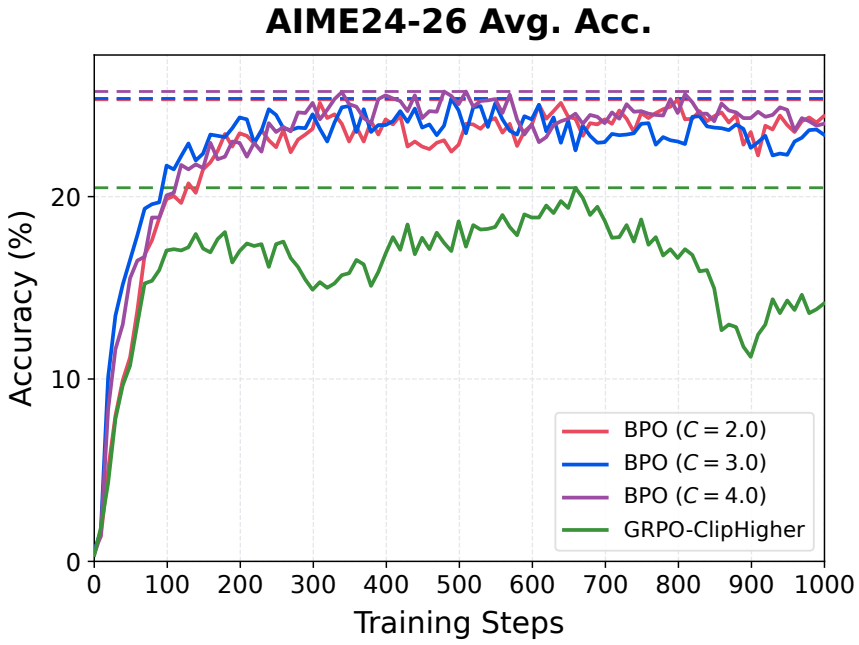

Figure 8: Effect of BPO's truncation parameter C on Qwen3-4B-Base with $\epsilon = 0.1$. Curves show mean Avg@32 across AIME24–26. Dashed lines mark the peak of each curve.

Table 8: Ablation on BPO's truncation parameter C with $\epsilon = 0.1$. Each entry is the peak Avg@32 (%) on the benchmark within the first 1000 training steps. The best result in each column is bold and the second best is underlined.

METHOD	AIME24	AIME25	AIME26
BPO ($C = 2.0$)	27.4	27.9	**25.1**
BPO ($C = 3.0$)	**28.4**	<u>28.0</u>	23.3
BPO ($C = 4.0$)	<u>27.6</u>	**29.0**	<u>24.0</u>
GRPO-ClipHigher	22.1	23.4	16.6

Across the three values of C, the peak accuracy of BPO ranges from 27.4% to 28.4% on AIME24, from 27.9% to 29.0% on AIME25, and from 23.3% to 25.1% on AIME26. All three settings exceed GRPO-ClipHigher on each benchmark. These results show that BPO is insensitive to C from 2.0 to 4.0.

H.3 Mismatch-Correction Weight

BPO w/ IS weight keeps the BPO clipping mask but replaces the truncated weight with the importance-sampling ratio. Its per-token loss is

$$\mathcal{L}_{i,t}(\pi) = -\hat{A}^i M_t^i(\omega) \mathrm{sg}\left(r_t^i\right) \log \pi(y_t^i | s_t^i).$$

Figure 9 shows the training curves. The two curves are close during the first 50 training steps, and the gap then widens. At step 199, the mean accuracy of BPO over AIME24–26 is 17.4 percentage points higher than that of BPO w/ IS weight. These results show that the mismatch-correction weight ω_t^i is essential to the performance of BPO, beyond the effect of the clipping mask.

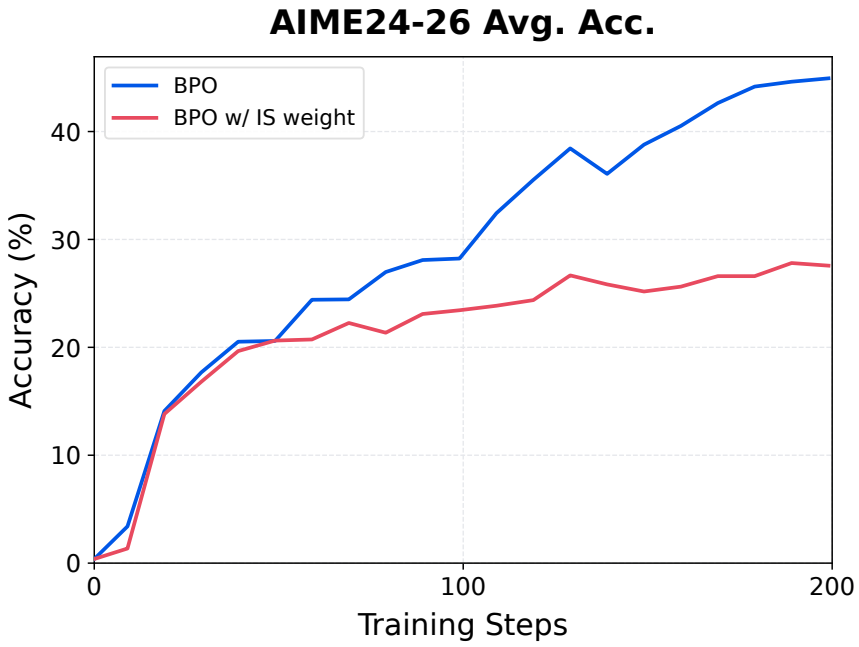

Figure 9: BPO versus BPO w/ IS weight on Qwen3-30B-A3B-Base. BPO w/ IS weight keeps the BPO clipping mask and uses the importance-sampling ratio as the weight. Curves show mean Avg@32 across AIME24–26.

H.4 Clipping Mask

We write $M_t^i(r)$ for the GRPO clipping mask in (30), which is computed from the importance-sampling ratio r_t^i. We write $M_t^i(\omega)$ for the BPO clipping mask in (31), which applies the same rule to ω_t^i. BPO w/ GRPO mask keeps the truncated weight of BPO but uses the GRPO clipping mask. Its per-token loss is

$$\mathcal{L}_{i,t}(\pi) = -\hat{A}^i M_t^i(r) \min\left\{\mathrm{sg}\left(\omega_t^i\right), C\right\} \log \pi(y_t^i|s_t^i).$$

Figure 10 shows the training curves. At step 199, the mean accuracy of BPO over AIME24–26 is 25.2 percentage points higher than that of BPO w/ GRPO mask. After step 79, the accuracy of BPO w/ GRPO mask stops improving and stays between 17.6% and 20.0%. These results show that the BPO weight ω_t^i should be used together with the BPO clipping mask $M_t^i(\omega)$.

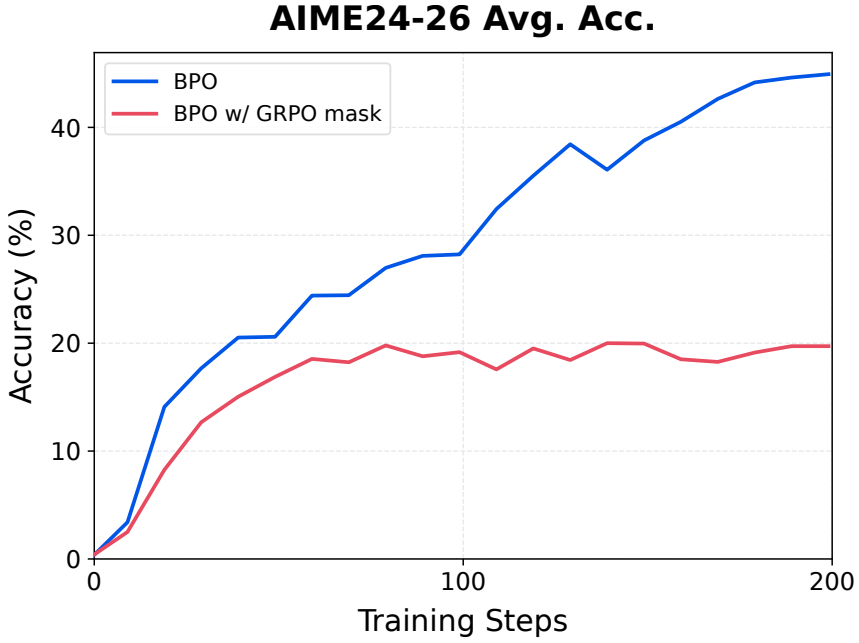

Figure 10: BPO versus BPO w/ GRPO mask on Qwen3-30B-A3B-Base. BPO w/ GRPO mask computes the clipping mask from the importance-sampling ratio. Curves show mean Avg@32 across AIME24–26.

H.5 Top-k KL Approximation

BPO replaces the full KL divergence in (14) with the binary KL divergence (17). We compare this choice with the top-k approximation (18) of the full KL divergence used in DPPO (Qi et al., 2026). BPO (Top-16 KL Approx) and BPO (Top-128 KL Approx) replace the KL divergence in (14) with its top-k approximation (18) with $k = 16$ and $k = 128$, respectively. Figure 11 shows the training curves, and Table 9 reports results at step 199.

BPO achieves 42.8% average accuracy, 3.0 percentage points higher than BPO (Top-128 KL Approx) and 5.9 points higher than BPO (Top-16 KL Approx). The binary approximation only needs the probability of the sampled token, while the top-k approximation needs the top-k probabilities at each position. All three methods are trained on 32 H200 GPUs. The wall-clock time to reach step 199 is 33.8 hours for BPO, compared with 56.8 hours for BPO (Top-16 KL Approx) and 152.6 hours for BPO (Top-128 KL Approx). BPO therefore achieves higher accuracy at a lower cost.

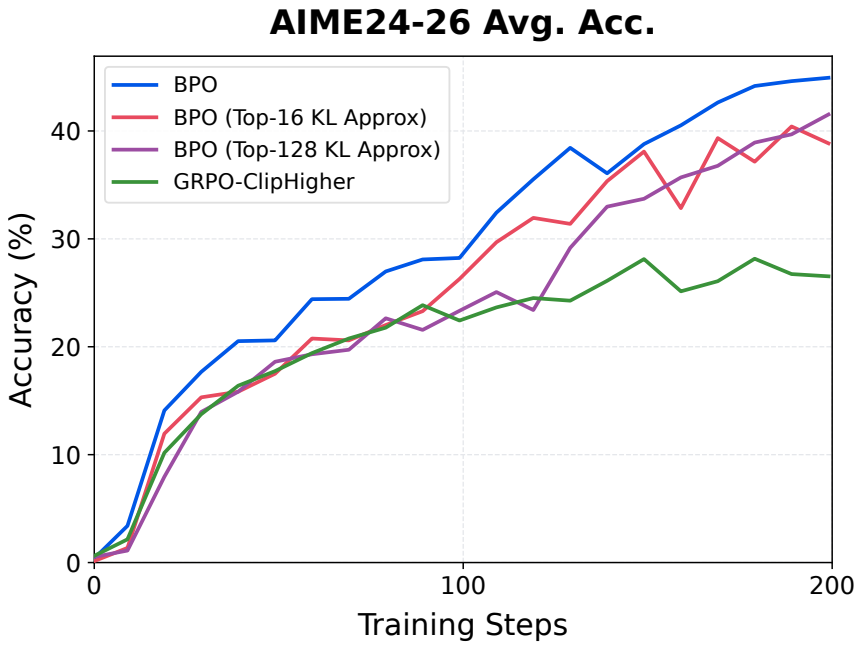

Figure 11: Binary KL approximation (BPO) versus top-k approximations of the full KL divergence on Qwen3-30B-A3B-Base. Curves show mean Avg@32 across AIME24–26.

Table 9: Avg@32 (%) of BPO with different KL approximations on Qwen3-30B-A3B-Base at training step 199. BPO (Top-k KL) denotes BPO (Top-k KL Approx) in Figure 11. The best result in each column is bold and the second best is underlined.

METHOD	AMC23	AIME24	AIME25	AIME26	HMMT25	HMMT26	AVG.
BPO	**93.5**	**53.2**	**35.8**	**45.7**	**16.6**	**11.8**	**42.8**
BPO (Top-16 KL)	84.7	45.7	31.9	39.0	11.3	8.6	36.9
BPO (Top-128 KL)	<u>89.8</u>	<u>47.6</u>	<u>34.4</u>	<u>42.6</u>	<u>13.2</u>	<u>11.3</u>	<u>39.8</u>
GRPO-ClipHigher	82.0	30.9	26.8	21.9	10.6	10.0	30.4