Bounded Ratio Reinforcement Learning

Yunke Ao * ETH Zurich
Le Chen † MPI for Intelligent Systems
Bruce D. Lee † ETH Zurich
Assefa S. Wahd ‡ University of Alberta
Aline Czarnobai ‡ Dartmouth College
Philipp Fürnstahl Balgrist University Hospital
Bernhard Schölkopf MPI for Intelligent Systems
Andreas Krause ETH Zurich

Abstract

Proximal Policy Optimization (PPO) has become the predominant algorithm for on-policy reinforcement learning due to its scalability and empirical robustness across domains. However, there is a significant disconnect between the underlying foundations of trust region methods and the heuristic clipped objective used in PPO. In this paper, we bridge this gap by introducing the *Bounded Ratio Reinforcement Learning (BRRL)* framework. We formulate a novel regularized and constrained policy optimization problem and derive its analytical optimal solution. We prove that this solution ensures monotonic performance improvement. To handle parameterized policy classes, we develop a policy optimization algorithm called *Bounded Policy Optimization (BPO)* that minimizes an advantage-weighted divergence between the policy and the analytic optimal solution from BRRL. We further establish a lower bound on the expected performance of the resulting policy in terms of the BPO loss function. Notably, our framework also provides a new theoretical lens to interpret the success of the PPO loss, and connects trust region policy optimization and the Cross-Entropy Method (CEM). We additionally extend BPO to *Group-relative BPO (GBPO)* for LLM fine-tuning. Empirical evaluations of BPO across MuJoCo, Atari, and complex IsaacLab environments (e.g., Humanoid locomotion), and of GBPO for LLM fine-tuning tasks, demonstrate that BPO and GBPO generally match or outperform PPO and GRPO in stability and final performance.

1 Introduction

Deep reinforcement learning (DRL) has achieved breakthroughs across diverse domains [27, 11, 17, 19]. Among DRL methods, Proximal Policy Optimization (PPO) [23] remains one of the most widely adopted algorithms. The core design of PPO is motivated by Trust Region Policy Optimization (TRPO, [21]), which constrains policy updates within a "trust region" to ensure stable iterations. By utilizing a first-order approximation of the TRPO objective, PPO achieves the scalability necessary for training modern large-scale models. As a result, PPO and its variant GRPO are now widely applied to tasks ranging from robotics to large language model (LLM) fine-tuning [13, 2, 26].

Despite its empirical success, PPO remains largely heuristic: its clipped objective is not directly derived from the trust-region formulation it was intended to approximate. Instead, the design of

*Correspondence to: yunke.ao@ai.ethz.ch
†Equal second author contribution
‡Equal third author contribution

Preprint.

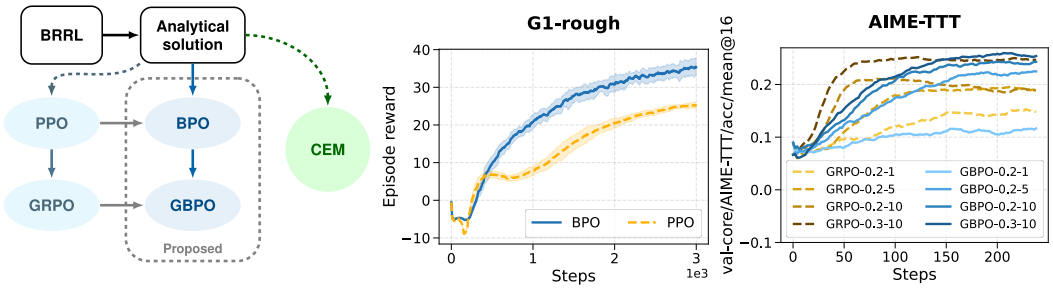

Figure 1: The Bounded Ratio Reinforcement Learning (BRRL) framework introduces the surrogate policy optimization problem under bounded ratio constraints. Its analytical solution closely relates to the PPO objective function, cross-entropy methods (CEM), and suggests a theoretically grounded policy optimization algorithm with minor changes to PPO: Bounded Policy Optimization (BPO). We observe marked improvements of BPO and its variant Group-relative BPO (GBPO) in the performance and stability for humanoid locomotion and LLM fine-tuning (mathematical reasoning).

the PPO objective was primarily driven by experimentation [23, 5]. Furthermore, most existing theoretical analyses of PPO's performance improvement rely on the original TRPO or policy gradient formulation [21, 12, 4], none of which fully capture the nuances of the first-order loss used in practice.

Numerous variants have been recently proposed to improve PPO. Some works focus primarily on algorithm design and report empirical performance gains without formal theoretical contributions [3, 34, 28, 6, 8]. Other works extend PPO to specific domains (e.g., safe RL, non-stationary RL) without modifying the core PPO loss function [1, 14]. There are also PPO variants aiming at improving the PPO loss from a theoretical lens [33, 30, 31, 18]. However, similar to PPO, they also utilize TRPO theory without introducing novel theoretical frameworks or establishing superior performance guarantees. Consequently, there remains a substantial gap between the theoretical foundations and the practical policy optimization algorithms.

To address this gap, we introduce the bounded ratio reinforcement learning (BRRL) framework. Instead of constraining policy updates through KL divergence [9] bounds as in TRPO, BRRL imposes bounded ratio constraints on the policy likelihood ratios. This formulation admits an analytic optimal policy, which reveals a simple structure for policy updates. We establish the following contributions using the BRRL framework:

- We derive the optimal solution of BRRL and prove its monotonic performance improvement guarantees. We also demonstrate that optimizing the PPO loss approximately pushes the policy towards this analytic optimal solution.
- We establish a connection between BRRL and the Cross-Entropy Method (CEM).
- We propose Bounded Policy Optimization (BPO), which optimizes an advantage-weighted divergence from the BRRL solution. We also extend BPO to Group-Relative BPO (GBPO), mirroring the extension from PPO to GRPO.
- We provide a performance improvement guarantee for the policy attained by BPO in terms of the loss that BPO optimizes.
- We demonstrate strong empirical performance of BPO on MuJoCo, Atari, IsaacLab locomotion tasks, and of GBPO for LLM fine-tuning.

Overall, BRRL provides a principled perspective on PPO-style algorithms, suggesting that their empirical success arises from approximating an analytically optimal bounded-ratio update. By more directly approximating this analytically optimal bounded-ratio update, BPO achieves improved empirical performance (Figure 1).

2 Notation

Markov Decision Process (MDP): We consider an infinite-horizon MDP defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r, d, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the transition model, $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the reward function, $d_0 : \mathcal{S} \to \mathbb{R}$ is the initial state distribution, and $\gamma \in (0, 1)$ is the discount factor. Let $\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ denote the stochastic policy.

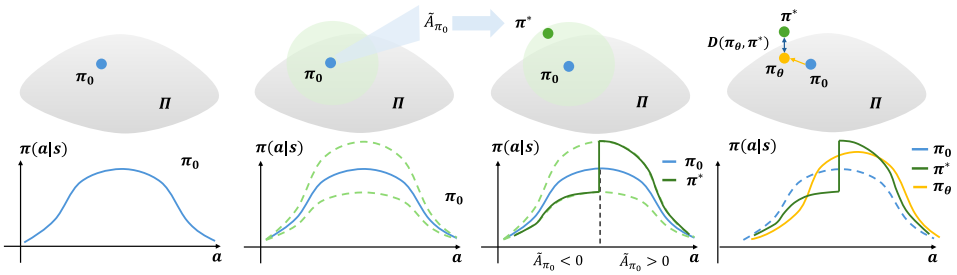

Figure 2: Illustration of Bounded Ratio RL (BRRL). (Left) Old policy π_0 within the parameterized policy class Π. (Middle Left) Construction of a trust region (light green) defined by bounded ratio constraints from Problem (4) or (9). (Middle Right) Estimation of the analytic optimal policy within the trust region (dark green, can be outside Π) using (soft-)median-advantages $\tilde{A}_{\pi_0}$ (Theorem 4.1). In general, actions with positive (resp. negative) advantages $\tilde{A}_{\pi_0} > 0$ (resp. $\tilde{A}_{\pi_0} < 0$) yield optimal ratios greater (resp. smaller) than 1. (Right) The updated policy within Π (yellow) is obtained by minimizing a divergence from the estimated optimal policy.

We denote $r_t := r(s_t, a_t, s_{t+1})$. The goal of the MDP is to solve the optimization problem

$$\max_{\pi \in \Pi} \eta(\pi) := \mathbb{E}_{s_{0:\infty}, a_{0:\infty}} \left[\sum_{t=0}^{\infty} \gamma^t r_t \right], \; s_0 \sim d_0, \; a_t \sim \pi(a_t|s_t), s_{t+1} \sim p(s_{t+1}|s_t, a_t), \tag{1}$$

which maximizes the expected discounted return under policy π within the policy class Π. Let us denote $d_\pi(s) := \sum_{t=0}^{\infty} \gamma^t P(s_t = s)$ as an unnormalized state visitation distribution [21]. Then the objective in (1) can be rewritten as $\eta(\pi) = \mathbb{E}_{s \sim d_\pi, a \sim \pi(\cdot|s)}[r_t]$.

Value function and advantage: We define the value function of a state s given policy π as $V_\pi(s) := \mathbb{E}_{s_0, a_0, \ldots | s_0 = s}[\sum_{t=0}^{\infty} \gamma^t r_t]$, and the Q-function $Q_\pi(s, a) := \mathbb{E}_{s_0, a_0, \ldots | s_0 = s, a_0 = a}[\sum_{t=0}^{\infty} \gamma^t r_t]$, where the actions (excluding the conditioned a_0 in the Q-function) are sampled from the policy π. The advantage function is defined as the difference between them $A_\pi(s, a) := Q_\pi(s, a) - V_\pi(s)$. As shown in [21], the expected return of the new policy π in (1) with regard to an old policy π_0 can be derived as

$$\eta(\pi) = \eta(\pi_0) + \mathbb{E}_{s \sim d_\pi, a \sim \pi(\cdot|s)}[A_{\pi_0}(s, a)]. \tag{2}$$

where the advantage is evaluated under π_0, and the expectation is taken over d_π and π.

Surrogate objectives: We denote the surrogate objective optimized by TRPO [21] as

$$L_{\pi_0}(\pi) := \eta(\pi_0) + \mathbb{E}_{s \sim d_{\pi_0}, a \sim \pi(\cdot|s)}[A_{\pi_0}(s, a)] = \eta(\pi_0) + \mathbb{E}_{s \sim d_{\pi_0}, a \sim \pi_0(\cdot|s)}[\rho A_{\pi_0}(s, a)], \tag{3}$$

where $\rho = \rho(a|s) := \pi(a|s)/\pi_0(a|s)$ are the importance weights. In contrast to (2), $L_{\pi_0}(\pi)$ takes the expectation over d_{π_0} instead of d_π. In TRPO, π is updated to optimize $L_{\pi_0}(\pi)$ with constrained KL-divergence from π_0.

3 Overview of Contributions

In this section, we present an overview of the contributions within this work, as shown in Figure 2.

Bounded ratio RL framework: We consider a policy optimization problem with bounded ratio trust region constraints from an old policy π_0, instead of the KL-divergence constraint of TRPO, as shown in Figure 2 (Middle Left). Specifically, with $L_{\pi_0}(\pi)$ defined in (3), the problem is expressed as

$$\max_{\pi} L_{\pi_0}(\pi), \quad \text{s.t. } 1 - \epsilon \leq \frac{\pi(a|s)}{\pi_0(a|s)} \leq 1 + \epsilon, \; \forall s, a. \tag{4}$$

Notably, this problem has an *analytical* optimal solution π^*, which in many cases (as detailed in Remark 4.3) can be derived as

$$\pi^*(a|s) = [1 + \epsilon \cdot \text{sign}(\tilde{A}_{\pi_0})] \cdot \pi_0(a|s), \tag{5}$$

where $\tilde{A}_{\pi_0} := Q_{\pi_0}(s, a) - \mu_{\pi_0}(s)$ is the median advantage. In particular, $\mu_{\pi_0}(s)$ denotes the median of $Q_{\pi_0}(s, a)$ over π_0, such that for any $s \in \mathcal{S}$, $\tilde{A}_{\pi_0}$ satisfies $\mathbb{E}_{a \sim \pi_0(\cdot|s)}[\text{sign}(\tilde{A}_{\pi_0})] = 0$. As

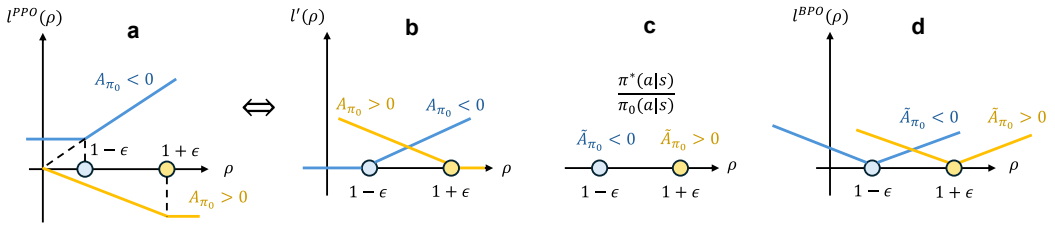

Figure 3: Loss functions of PPO and bounded-ratio RL. Curves for $\tilde{A}_{\pi_0} > 0$ and $\tilde{A}_{\pi_0} < 0$ are shown in yellow and blue, respectively. (a) Original PPO loss function. (b) Equivalent loss function of PPO as introduced in Proposition 4.6. (c) Optimal ratios for the optimization problem with bounded ratio constraints in (4). (d) Advantage-weighted TV loss function in BPO, defined in (8).

shown in Figure 2 (Middle Right) and Figure 3 (c), this optimal solution can be explained as: if $Q_{\pi_0}(s,a)$ is higher than the threshold $\mu_{\pi_0}(s)$, then take the highest probability within the constraint $\pi^*(a|s) = (1+\epsilon)\pi_0(a|s)$; otherwise, let $\pi^*(a|s) = (1-\epsilon)\pi_0(a|s)$. Threshold $\mu_{\pi_0}(s)$ is selected as the median, s.t. π^* is a normalized probability distribution ($\sum_a \pi^*(a|s) = 1$). A formal theorem on the optimal solution for general cases is provided in Theorem 4.1. Note that Theorem 4.1 can also be extended to problems with asymmetric bounded ratio constraints ($c_l \leq \pi(a|s)/\pi_0(a|s) \leq c_h$). This asymmetric solution is used to draw a connection to the cross-entropy method (CEM) in Section 4.6.

Monotonic performance guarantee: For the cases where optimal policy π^* from (5) is optimal, it can be shown to have improved performance over π_0

$$\eta(\pi^*) = \eta(\pi_0) + \epsilon\, \mathbb{E}_{s\sim d_{\pi^*}, a\sim\pi_0}[\text{sign}(\tilde{A}_{\pi_0})\tilde{A}_{\pi_0}] := \eta(\pi_0) + \epsilon B, \tag{6}$$

where the second term is non-negative and is positive whenever π_0 induces non-zero median advantage. For a fixed π_0, we denote this constant improvement term as ϵB. A performance bound for general cases is provided in Theorem 4.2. Though π^* is simple to express and provides improvement guarantees, it may not lie in the admissible policy class Π (Figure 2 Middle right). This motivates the design of policy optimization algorithms to minimize divergence between the policy $\pi \in \Pi$ and π^*.

Revisiting the PPO loss function: We observe that the PPO loss function approximately drives the policy towards π^* in (5). Specifically, as shown in Figure 3 (a-b), optimizing the PPO objective [23] is equivalent to minimizing the expectation of the following loss function evaluated at $\rho = \pi(a|s)/\pi_0(a|s)$

$$l'(\rho) := \begin{cases} |A_{\pi_0}| \cdot |\rho - (1 + \epsilon \cdot \text{sign}(A_{\pi_0}))|, & |\rho - 1| \leq \epsilon, \\ 0, & |\rho - 1| > \epsilon. \end{cases} \tag{7}$$

A formal theorem on this equivalence with step-by-step proof is detailed in Section 4.4 and Appendix A.5. At the beginning of the iteration, the ratio always starts from 1, and the PPO loss minimizes an *advantage-weighted absolute error* between the ratio ρ and the target $1 + \epsilon\text{sign}(A_{\pi_0})$, then it applies zero-gradient after reaching the target. Note that this target ratio closely matches the solution in (5), except that PPO uses the mean advantage A_{π_0}, and the BRRL solution is expressed in terms of the median advantage $\tilde{A}_{\pi_0}$.

Bounded Policy Optimization (BPO): Building on the solution in (5), we introduce a natural PPO variant with the loss function l^{BPO} to directly minimize the *advantage weighted total variation* from the optimal solution. For the solution in (5), the loss l^{BPO} evaluated under $\rho = \pi(a|s)/\pi_0(a|s)$ is

$$l^{BPO}(\rho) := |A_{\pi_0}| \cdot \left|\rho - \frac{\pi^*(a|s)}{\pi_0(a|s)}\right| = |A_{\pi_0}| \cdot |\rho - (1 + \epsilon \cdot \text{sign}(\tilde{A}_{\pi_0}))|. \tag{8}$$

The loss is illustrated in Figure 3 (d). Compared with the PPO loss in (7), this loss function l' only differs in two ways: (1) a symmetric slope also for $|\rho - 1| \geq \epsilon$ and (2) using $\tilde{A}_{\pi_0}$ instead of A_{π_0}. In practice, this also requires learning an additional median value function alongside the mean value function, though the median can be approximated by the mean to reduce computational overhead. Notably, with this refined loss function, BPO has both *theoretical performance guarantees* (discussed below) and strong empirical performance, as demonstrated in Section 5. The same loss function can also be adapted for LLM fine-tuning, analogous to how PPO was adapted to GRPO (Section 4.5).

BPO performance guarantees: Assuming the optimal solution in (5) is valid, we can express the stepwise improvement in terms of the achieved loss (8). Specifically, we show that

$$\eta(\pi) \geq \eta(\pi_0) + \epsilon B - \mathbb{E}_{s\sim d_{\pi_0}, a\sim\pi_0}\left[l^{BPO}\left(\frac{\pi(a|s)}{\pi_0(a|s)}\right)\right] - \delta(\pi, \pi^*),$$

where B is defined in (6). Here, $\delta(\pi, \pi^*)$ is an error term that is related to $l^{BPO}(\frac{\pi(a|s)}{\pi_0(a|s)})$ and reduces to 0 if we have perfect policy approximation $\pi = \pi^*$. This theoretical result directly implies that, if our loss function l^{BPO} is sufficiently minimized over states and actions sampled from π_0, and if the policy approximation error is small, we can obtain monotonic performance improvement. The formal result is detailed in Corollary 4.5.

4 Method

We now proceed to present the aforementioned framework of BRRL and its extensions.

4.1 Bounded Ratio RL Framework

Intuitively, for an MDP with finite state and action spaces, Problem (4) is a linear programming problem. Specifically, for a fixed state s, the optimization variable $\pi(a|s)$ is a finite-dimensional vector. Consequently, the objective function and constraints in Problem (4) are linear in $\pi(a|s)$. However, for general state and action spaces, the optimal solution of this linear programming problem is difficult to specify analytically. Nevertheless, an additional *regularizer* allows for the derivation of the general analytical solution. Namely, we consider the following regularized constrained optimization problem:

$$\begin{aligned} &\max_{\pi}\ L_{\pi_0}(\pi) - \lambda\mathbb{E}_{s\sim d_{\pi_0}, a\sim\pi_0}\left[H\left(\frac{\pi(a|s)}{\pi_0(a|s)}\right)\right], \\ &\text{where}\quad H(\rho) := (\rho - 1 + \epsilon)\log(\rho - 1 + \epsilon) + (1 + \epsilon - \rho)\log(1 + \epsilon - \rho). \end{aligned} \tag{9}$$

Here the regularizer $H(\rho) \in [2\epsilon\log\epsilon, 0)$ decreases as $\rho \to 1$ and increases as $\rho \to 1 \pm \epsilon$. Moreover, its gradient becomes unbounded near the boundaries $1 \pm \epsilon$, so H provides log barriers for the original bounded ratio constraints $1 - \epsilon < \frac{\pi(a|s)}{\pi_0(a|s)} < 1 + \epsilon$. The regularizer is weighted by λ. According to Fermi-Dirac statistics [10], Problem (9) has a closed-form solution, detailed in the following theorem.

Theorem 4.1 (Optimal solution)**.** *The optimal policy π^* of the problem described in* (9) *satisfies:*

$$\pi^*(a|s) = \left(1 + \epsilon\tanh\left(\frac{\tilde{A}_{\pi_0}}{2\lambda}\right)\right)\pi_0(a|s),\ \ \tilde{A}_{\pi_0} := Q_{\pi_0}(s,a) - \mu_{\pi_0}(s),$$

where $\tilde{A}_{\pi_0}$ is the soft-median advantage and $\mu_{\pi_0}(s)$ is the soft-median of $Q_{\pi_0}(s,a)$ that satisfies

$$\mathbb{E}_{a\sim\pi_0(\cdot|s)}\left[\tanh\left(\frac{\tilde{A}_{\pi_0}}{2\lambda}\right)\right] = 0 \ \Leftrightarrow\ \mu_{\pi_0}(s) = \arg\min_{\mu(s)}\mathbb{E}_{a\sim\pi_0(\cdot|s)}\left[g\left(\frac{Q_{\pi_0}(s,a) - \mu(s)}{\lambda}\right)\right],$$

where $g : \mathbb{R} \to \mathbb{R}_{\geq 0}, g(x) = \ln(e^{-\frac{x}{2}} + e^{\frac{x}{2}})$ is a soft absolute function.

The detailed proof of Theorem 4.1 is provided in Appendix A.1. Intuitively, the optimal solution assigns a higher ratio to actions with a higher advantage while keeping the ratio between $[1-\epsilon, 1+\epsilon]$.

We can obtain a monotonic performance guarantee for the optimal solution.

Theorem 4.2 (Monotonic performance guarantee)**.** *The optimal policy in Theorem 4.1 satisfies*

$$\eta(\pi^*) \geq \eta(\pi_0) + \epsilon\mathbb{E}_{s\sim d_{\pi^*}, a\sim\pi_0(\cdot|s)}\left[\tanh\left(\frac{\tilde{A}_{\pi_0}}{2\lambda}\right)\tilde{A}_{\pi_0}\right] =: \eta(\pi_0) + \epsilon B,$$

where $\tilde{A}_{\pi_0}$ abbreviates $\tilde{A}_{\pi_0}(s,a)$, B is a non-negative constant given fixed π_0.

The proof of Theorem 4.2 is detailed in Appendix A.2. Note that the term $\tanh(\frac{\tilde{A}_{\pi_0}}{2\lambda})\tilde{A}_{\pi_0}$ is always non-negative since the signs of $\tanh(\frac{\tilde{A}_{\pi_0}}{2\lambda})$ and $\tilde{A}_{\pi_0}$ are always the same. Therefore, our optimal policy π^* guarantees monotonic improvement with an analytical improvement bound, in contrast to the guarantee for TRPO in Theorem 1 of [21]. However, the policy π^* may not be a member of the class of parameterized policies Π. Consequently, in Section 4.2 and 4.3, we further develop the policy optimization loss algorithm by minimizing a certain divergence from π to π^*.

Remark 4.3 (Optimal solution to unregularized Problem (4)). Note that by taking $\lambda \to 0$ in Theorem 4.1 and Theorem 4.2, one can obtain the optimal ratio and monotonic guarantees for the unregularized Problem (4). In many cases, one can simplify the resulting optimal policy as $\pi^*(a|s) = [1 + \epsilon \text{sign}(\tilde{A}_{\pi_0})]\pi_0(a|s)$ in (5), where $\mu_{\pi_0}(s)$ is the median of $Q_{\pi_0}(s,a)$ over $\pi_0(\cdot|s)$. Such simplification holds if $\forall\, s,\ \exists\, \mu(s)$, such that

$$\mathbb{E}_{a\sim\pi_0(\cdot|s)}[\text{sign}(Q_{\pi_0}(s,a) - \mu(s))] = 0. \tag{10}$$

Otherwise, the simplified π^* can never be normalized. One valid case is a uniform density $\pi_0(\cdot|s)$ with continuous $\mathcal{A}$ and a Q-function $Q_{\pi_0}(s,a)$ which is smooth over a. However, there are also counterexamples. Consider, for instance, an MDP with a single state and a discrete action space $\mathcal{A} = \{a_1, a_2\}$. Assume that $Q_{\pi_0}(a_2) > Q_{\pi_0}(a_1)$, and $\pi_0(a_1) = \frac{1}{4}, \pi_0(a_2) = \frac{3}{4}$. Then for any $\mu \in \mathbb{R}$, condition (10) does not hold. While the simplified interpretation of π^* is only valid in special cases, the result of Theorem 4.1 still holds for arbitrarily small $\lambda > 0$ and general spaces (see Appendix A.1).

4.2 Alternative Perspective: Minimizing Divergence from Optimal Policy

In this section, we consider the policy optimization problem as minimizing the divergence to the optimal solution, instead of directly applying policy gradient methods. Specifically, given the optimal policy obtained from Theorem 4.1, we can formulate policy optimization as

$$\min_{\pi_\theta\in\Pi} D(\pi_\theta, \pi^*),$$

where π_θ is the parameterized policy, D is a divergence function such as the KL-divergence, total variation (TV), etc. Specifically, the TV for each state can be expressed as

$$D_\theta^{TV}(s) := \sum_a |\pi^*(a|s) - \pi_\theta(a|s)| = \mathbb{E}_{a\sim\pi_0(\cdot|s)}\left[\left|\frac{\pi^*(a|s)}{\pi_0(a|s)} - \frac{\pi_\theta(a|s)}{\pi_0(a|s)}\right|\right]. \tag{11}$$

We also consider an advantage-weighted TV (ATV) loss function defined as

$$D_\theta^{ATV}(s) := \mathbb{E}_{a\sim\pi_0(\cdot|s)}\left[\left|\frac{\pi^*(a|s)}{\pi_0(a|s)} - \frac{\pi_\theta(a|s)}{\pi_0(a|s)}\right| \cdot |A_{\pi_0}|\right]. \tag{12}$$

Notably, this divergence is directly correlated with the performance improvement of the parameterized policy π_θ, as detailed in the following Corollary.

Corollary 4.4 (Performance improvement guarantee with policy approximation error)**.** *Consider D_θ^{ATV} defined in* (12) *with π^* given from Theorem 4.1. Then it holds that*

$$\eta(\pi_\theta) \geq \eta(\pi_0) + \mathbb{E}_{s\sim d_{\pi_\theta}, a\sim\pi_0(\cdot|s)}\left[\epsilon \tanh\left(\frac{\tilde{A}_{\pi_0}}{2\lambda}\right)\tilde{A}_{\pi_0} - D_\theta^{ATV}(s)\right],$$

where $\tilde{A}_{\pi_0}$ abbreviates $\tilde{A}_{\pi_0}(s,a)$.

The proof of Corollary 4.4 is provided in A.3. Corollary 4.4 shows that by minimizing the loss D_θ^{ATV} over the state distribution d_{π_θ}, we can improve performance w.r.t. policy π_θ as long as the parameterized policy class is sufficiently expressive. However, minimizing $\mathbb{E}_{s\sim d_{\pi_\theta}}$ over the policy parameters θ is non-trivial due to the dependence of the state distribution on π_θ. Consequently, in practice we only optimize the expectation of the divergence over the old policy, leading to the objectives

$$J^{ATV}(\theta) := \mathbb{E}_{s\sim d_{\pi_0}}[D_\theta^{ATV}(s)], \quad J^{TV}(\theta) := \mathbb{E}_{s\sim d_{\pi_0}}[D_\theta^{TV}(s)]. \tag{13}$$

The following Corollary 4.5 expresses a lower bound on the performance of the policy π_θ in terms of these quantities.

Corollary 4.5 (Performance improvement guarantee with loss functions)**.** *With B defined in Theorem 4.2, and $\tilde{\delta} := \max_s \mathbb{E}_{a\sim\pi_0(\cdot|s)}\left[\tanh\left(\frac{\tilde{A}_{\pi_0}}{2\lambda}\right)\tilde{A}_{\pi_0}\right]$, it holds that*

$$\eta(\pi_\theta) \geq \eta(\pi_0) + \epsilon B - J^{ATV}(\theta) - \frac{D^{ATV}_{\max}}{1-\gamma} J^{TV}(\theta) - \frac{\gamma\epsilon D^{ATV}_{\max}}{(1-\gamma)^2} - \frac{\gamma\epsilon\tilde{\delta} D^{TV}_{\max}}{(1-\gamma)^2},$$

where $D^{ATV}_{\max} := \max_s D^{ATV}_\theta(s)$ and $D^{TV}_{\max} := \max_s D^{TV}_\theta(s)$.

The proof of Corollary 4.5 is detailed in Appendix A.4. Corollary 4.5 decomposes the performance lower bound into a non-negative term, ϵB, and several negative terms which depend on the gap between the optimized policy π_θ and the policy π^*. The first two of these gap terms depend on $J^{ATV}(\theta)$ and $J^{TV}(\theta)$. Both of these quantities can be estimated from trajectories collected under π_0 and minimized by optimizing θ, motivating a loss defined as a weighted combination of these terms. The other two gap terms are characterized by the worst case divergence of π_θ from π^* over the state space through the quantities $D^{ATV}_{\max}$ and $D^{TV}_{\max}$. Though these quantities are generally not computable, they can be related to the expected losses over the distribution d_{π_0} under additional assumptions (e.g., adequate state coverage under d_{π_0}). Notably, with perfect policy approximation ($\pi_\theta = \pi^*$), it holds that $J^{ATV}(\theta) = J^{TV}(\theta) = D^{ATV}_{\max} = D^{TV}_{\max} = 0$, recovering the original monotonic performance guarantee of Theorem 4.2.

In contrast, the bound from TRPO in Theorem 1 of [21] does not contain a positive term. Besides, it penalizes the worst-case divergence between the updated policy and the current policy π_0 through $D^{\max}_{TV}(\pi, \pi_0)$, rather than the approximation error to an ideal solution π^*. Consequently, this negative term reflects the *magnitude of the update* away from π_0. It vanishes only in the degenerate case $\pi = \pi_0$ (i.e., no policy change), and is generally nonzero whenever a nontrivial policy update occurs.

Corollary 4.5 also motivates choosing a small ϵ. For small $\epsilon > 0$, π^* remains close to π_0 by construction, so matching π^* typically requires only a small deviation from a realizable policy in the class (namely π_0), making the approximation error terms easier to control. Since π_0 is realizable in the policy class, the optimal values of $J^{ATV}(\theta)$ and $J^{TV}(\theta)$ approach zero as $\epsilon \to 0$. Therefore, under such regularity conditions, it holds that $\frac{\gamma D^{ATV}_{\max}}{(1-\gamma)^2} + \frac{\gamma\tilde{\delta} D^{TV}_{\max}}{(1-\gamma)^2} \ll B$ for ϵ sufficiently small, therefore guaranteeing improvement.

4.3 Bounded Policy Optimization

In this section, we present the practical implementation of our algorithm. As in PPO, we use a value network V_ϕ to estimate A_{π_0}. Specifically, we estimate the return value $R_\phi(s,a)$ using generalized advantage estimation [22], and use it to update the value function by minimizing

$$J^{VF}(\phi) := \mathbb{E}_{s\sim d_{\pi_0}, a\sim\pi_0(\cdot|s)}[(R_\phi(s,a) - V_\phi(s))^2]. \tag{14}$$

In addition, following Theorem 4.1, we further train a network μ_ψ to minimize the normalization loss

$$J^{MF}(\psi) := \mathbb{E}_{s\sim d_{\pi_0}, a\sim\pi_0(\cdot|s)}\left[\lambda g\left(\frac{R_\phi(s,a) - \mu_\psi(s)}{\lambda}\right)\right]. \tag{15}$$

The practical loss function for θ uses the estimated advantage function to approximate $J^P(\theta) := J^{ATV}(\theta) + \alpha_1 J^{TV}(\theta)$ with α_1 as a tunable weight:

$$\hat{J}^P(\theta) := \mathbb{E}_{s\sim d_{\pi_0}, a\sim\pi_0(\cdot|s)}\left[\left|1 + \epsilon\tanh\left(\frac{\hat{A}_{\pi_0}}{2\lambda}\right) - \frac{\pi_\theta(a|s)}{\pi_0(a|s)}\right| \cdot (|R_\phi(s,a) - V_\phi(s)| + \alpha_1)\right], \tag{16}$$

where $\hat{A}_{\pi_0} := R_\phi(s,a) - \mu_\psi(s)$. Note that $\hat{J}^P(\theta)$ is not exactly J^P, but the gap can be controlled by minimizing the estimation error of V_ϕ and μ_ψ. Our final bounded policy optimization algorithm follows a PPO-style training procedure, summarized in Algorithm 1.

4.4 Revisiting the PPO Objective

In this section, we connect our theory and algorithmic framework to PPO [23]. In PPO, the following surrogate objective function is introduced

$$\mathbb{E}_{s\sim d_{\pi_0}, a\sim\pi_0}\left[\min\left\{\text{clip}\left(\rho, 1-\epsilon, 1+\epsilon\right) A_{\pi_0}, \rho A_{\pi_0}\right\}\right] := J_{PPO}(\theta), \tag{17}$$

Algorithm 1 Bounded policy optimization (BPO)

1: Initialize $\pi_\theta, V_\phi, \mu_\psi$, choose a sufficiently small λ
2: **for** $i = 1, 2, ...$ **do**
3: Assign $\pi_0 \leftarrow \pi_\theta$
4: Run π_0 for N steps, and collect the dataset $\mathcal{D} := \{s^j, a^j, R^j, \pi_0(a^j|s^j)\}_{j=1}^N$.
5: Update θ, ϕ, ψ by minimizing

$$\hat{J}^P(\theta) + w_1 J^{VF}(\phi) + w_2 J^{MF}(\psi),$$

where $\hat{J}^P, J^{VF}, J^{MF}$ are defined in (16), (14), (15), and evaluated from $\mathcal{D}$.
6: **end for**

where $\rho := \frac{\pi_\theta(a|s)}{\pi_0(a|s)}$ is the ratio between the new and old policies, and A_{π_0} denotes $A_{\pi_0}(s, a)$.

We observe a strong correlation between the BPO loss function and the PPO loss function. To show this correlation, we first introduce an equivalent form of the PPO loss in the following proposition.

Proposition 4.6. *Optimizing the loss function $J_{PPO}(\theta)$ in* (17) *is equivalent to minimizing the following function*

$$J'(\theta) := \mathbb{E}_{s\sim d_{\pi_0}, a\sim\pi_0(\cdot|s)} \left[l' \left(\frac{\pi_\theta(a|s)}{\pi_0(a|s)} \right) \right],$$

where

$$l'(\rho) := \begin{cases} |A_{\pi_0}| \cdot |\rho - (1 + \epsilon \cdot \textit{sign}(A_{\pi_0}))|, & |\rho - 1| \le \epsilon, \\ 0, & |\rho - 1| > \epsilon. \end{cases}$$

Intuitively, this equivalence follows from the fact that adding or subtracting a constant from the objective function does not change the optimal solution. This is illustrated in Fig. 3 (a vs b). A proof is provided in Appendix A.5.

On the other hand, as $\lambda \to 0$, the loss J^{ATV} in (13) can, in many cases (Remark 4.3), be expressed as

$$J^{ATV}(\theta) \approx \mathbb{E}_{s\sim d_{\pi_0}, a\sim\pi(\cdot|s)}[l^{BPO}(\rho)], \quad \text{where } l^{BPO}(\rho) = |A_{\pi_0}| \cdot |\rho - (1 + \epsilon \cdot \text{sign}(\tilde{A}_{\pi_0}))|.$$

Thus, the loss l^{BPO} resembles the PPO loss l' in Proposition 4.6 when $|\rho - 1| \le \epsilon$, as detailed in Section 3. For $|\rho - 1| > \epsilon$, BPO penalizes the policy for deviating from the original policy, which encourages satisfaction of the bounded-ratio constraints. This is also partially addressed by the zero gradient of PPO and the target KL divergence mechanism [25], which slows down the update of the new policy if it deviates too far from the original policy (i.e., if the KL divergence between the two surpasses the target KL divergence). Recent PPO variants also utilize similar ideas by introducing negative gradients when $|\rho - 1| > \epsilon$ [30, 33], which can be theoretically justified by our framework. Although the exact optimization dynamics of BPO and PPO differ, both follow a common principle: drive the policy ratio from 1 toward the (approximate) analytical optimum of BRRL and then stop. This offers a key insight into the underlying success of PPO-based methods.

4.5 Extension to LLM Fine-Tuning

In the context of LLM fine-tuning, training an additional critic can be computationally expensive. This challenge motivates the design of Group Relative Policy Optimization (GRPO) [26], which estimates advantages relative to a group of concurrent samples rather than utilizing an auxiliary value network. Building on this idea, we introduce Group-relative Bounded Policy Optimization (GBPO), an extension of BPO derived from Theorem 4.1. Specifically, for a given prompt q, the model generates a group of sampled outcomes $\{o_1, o_2, \ldots, o_G\}$. A reward model then assigns a score to each output, denoted by $\mathbf{R} = \{r_1, r_2, \ldots, r_G\}$. As in standard GRPO, we estimate advantages using z-scores $A_i := \frac{r_i - \text{mean}(\mathbf{R})}{\text{std}(\mathbf{R})}$. As noted in Remark 4.3, when the regularization parameter λ is small, the implicit baseline $\mu_{\pi_0}(q)$ converges to the median of the Q-values. We therefore also estimate the

median-advantage as $\tilde{A}_i := \frac{r_i - \text{median}(\mathbf{R})}{\text{std}(\mathbf{R})}$. The GBPO objective function is then defined as:

$$\hat{J}^P(\theta) = \mathbb{E}_{\substack{q\sim P(\mathcal{Q}) \\ o_i\sim\pi_0(\cdot|q)}} \left[\frac{1}{G}\sum_{i=1}^{G} \frac{1}{|o_i|}\sum_{t=1}^{|o_i|} \left| 1 + \epsilon \tanh\left(\frac{\tilde{A}_i}{2\lambda}\right) - \frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_0(o_{i,t}|q,o_{i,<t})} \right| \cdot |A_{i,t}| \right],$$

where t denotes the token index, and $\mathcal{Q}$ is the question set. In scenarios where a reward is only provided at the end of the sequence, the step-dependent advantages $A_{i,t}$ and $\tilde{A}_{i,t}$ are equal to the sequence-level A_i and $\tilde{A}_i$, respectively. If per-step scores are available, these advantages can be estimated token-wise following the approach in [26].

4.6 Asymmetric Ratio Constraints and Cross Entropy Method

In this section, we generalize Theorem 4.1 to asymmetric ratio constraints. Similar to (9), we consider the regularized problem with general ratio boundaries $\forall s, a,\ c_l \le \frac{\pi(a|s)}{\pi_0(a|s)} \le c_h$, with $c_l < 1 < c_h$

$$\begin{aligned} &\max_{\pi}\ L_{\pi_0}(\pi) - \lambda \mathbb{E}_{s\sim d_{\pi_0}, a\sim\pi_0}\left[H'\left(\frac{\pi(a|s)}{\pi_0(a|s)}\right)\right], \\ &\text{where}\quad H'(\rho) := (\rho - c_l)\log(\rho - c_l) + (c_h - \rho)\log(c_h - \rho) + \log\frac{c_h - 1}{1 - c_l}\rho. \end{aligned} \tag{18}$$

Here, the regularizer H' still takes its maximum at $\rho = 1$, and provides log barriers for the asymmetric constraints $c_l \le \rho \le c_h$. The optimal solution is detailed in the following Corollary.

Corollary 4.7. *(Asymmetric optimal policy) The optimal policy π^* of the problem* (18) *satisfies*

$$\frac{\pi^*(a|s)}{\pi_0(a|s)} = c_l + \frac{c_h - c_l}{1 + \frac{c_h - 1}{1 - c_l}\exp(-\tilde{A}'_{\pi_0}/\lambda)},\ \tilde{A}'_{\pi_0} := Q_{\pi_0}(s,a) - \mu'_{\pi_0}(s),$$

where $\mu'_{\pi_0}(s)$ is the soft-$\frac{c_h-1}{c_h-c_l}$-quantile satisfying

$$\mu'_{\pi_0}(s) = \arg\min_{\mu(s)} \mathbb{E}_{a\sim\pi_0(\cdot|s)}\left[g'\left(\frac{Q_{\pi_0}(s,a) - \mu(s)}{\lambda}\right)\right],$$

where $g' : \mathbb{R} \to \mathbb{R}_{\ge 0}, g'(x) = \ln\left(e^{\frac{c_h-1}{c_h-c_l}x} + \frac{c_h-1}{1-c_l} e^{-\frac{1-c_l}{c_h-c_l}x}\right)$.

Similar to Theorem 4.1, these results are closely related to policy optimization with asymmetric clip ratios [29, 32]. A monotonic performance guarantee similar to Theorem 4.2 is provided in Appendix A.7. Moreover, when $\lambda \to 0$, we also have $\mu'_{\pi_0}(s)$ the exact $\frac{c_h-1}{c_h-c_l}$-quantile in many cases, similar to Remark 4.3. Notably, when $c_l = 0$, $\lambda \to 0$, and the $\frac{c_h-1}{c_h-c_l}$-quantile exists for $Q_{\pi_0}(s,a)$, we have $\pi^*(a|s) = c_h\pi_0(a|s)$ for $Q_{\pi_0}(s,a) > \mu'_{\pi_0}(s)$ and 0 otherwise. This recovers a cross-entropy method (CEM) when π_0 is uniform, where the optimal solution at each iteration assigns non-zero probability to the top $\frac{1-c_l}{c_h-c_l}$ samples.

5 Experiments

In this section, we present extensive experiments to validate the proposed BPO algorithm. We first benchmark its performance against PPO across standard MuJoCo and Atari environments (Section 5.1). To assess scalability, we evaluate BPO within NVIDIA IsaacLab [15], a high-throughput simulation platform capable of simulating *thousands of* parallel environments for real-world robotic policy training. Furthermore, we apply our GBPO variant to LLM fine-tuning tasks, and compare it directly against GRPO (Section 5.3). Then, we dive deeper into the analysis of the ratio statistics during training, connecting it to the performance gap between BPO and PPO. Finally, we conduct an ablation study to analyze the sensitivity of training performance to key components, including the loss function, the λ parameter, and various loss coefficients. All hyperparameters are detailed in Appendix A.8

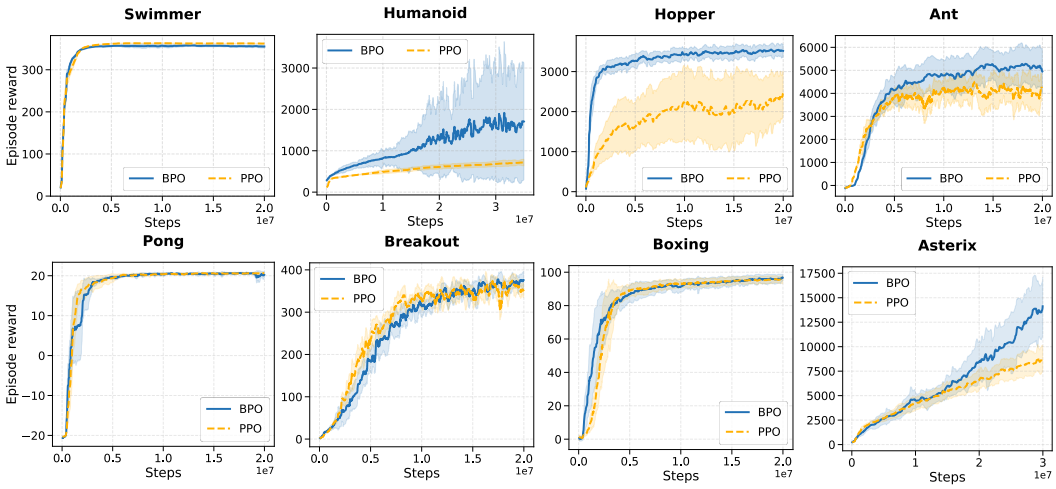

Figure 4: BPO versus PPO on MuJoCo and Atari environments. Shaded regions represent the standard deviation across 10 random seeds. In most environments, BPO matches or outperforms PPO.

5.1 Benchmarking with Classical Environments

We compare the performance of BPO and PPO in classical environments. For these experiments, BPO was implemented within the Stable Baselines3 framework, with hyperparameters for all baseline algorithms sourced from RL-Zoo [20]. As shown in Figure 4, BPO performs competitively with or superior to PPO across a range of classical benchmarks. Specifically, in MuJoCo tasks, BPO achieves clear performance gains in the Ant-v4, Hopper-v4, and Humanoid-v4 environments. Training on Humanoid-v4 exhibits high variance for BPO, characterized by significant performance divergence across random seeds. Both PPO and BPO struggle to achieve peak performance in this environment, primarily due to limited sample efficiency. However, as demonstrated in Section 5.2, both methods successfully solve more complex humanoid tasks when provided with sufficient samples. In Atari benchmarks, BPO generally matches PPO's performance, notably outperforming it in the Asterix environment.

We report our benchmarking results against off-policy baselines in MuJoCo and Atari environments in Table 1. While SAC [7] outperforms both PPO and BPO in the Ant-v4 and Humanoid-v4 tasks, it fails to achieve competitive results in Swimmer-v4. In contrast, BPO consistently outperforms PPO in the Ant-v4, Humanoid-v4, and Hopper-v4 environments while remaining competitive in Swimmer-v4. Both BPO and PPO consistently outperform DQN [16] in Atari benchmarks.

Table 1: Comparison of converged total rewards between BPO, PPO, and off-policy algorithms. Bolded and underlined numbers indicate the highest and second-highest results across all tested algorithms. For AsterixNoFrameskip-v4, the algorithms are evaluated after the same wall-clock time (12h). Rewards in other environments are evaluated after convergence.

Mujoco Envs	BPO	PPO	SAC	Atari Envs	BPO	PPO	DQN
Ant-v4	4871.4	4230.1	**6161.8**	BreakoutNoFrameskip-v4	**374.6**	360.4	252.5
Humanoid-v4	1570.4	781.3	**6806.4**	PongNoFrameskip-v4	**20.6**	**20.6**	**20.6**
Hopper-v4	**3505.1**	2497.7	3015.1	BoxingNoFrameskip-v4	94.7	**95.7**	92.5
Swimmer-v4	354.6	**362.4**	102.7	AsterixNoFrameskip-v4	**11247.9**	9471.5	7122.8

5.2 Benchmarking with IsaacLab Environments

In this section, we evaluate the scalability and performance of BPO relative to PPO within the IsaacLab simulation platform. We focus on four challenging locomotion tasks on rough terrain: Go1-rough, Anymal-C-rough, G1-rough, and H1-rough, which require the agents (quadrupeds like Unitree Go1 and Anymal-C or humanoids like Unitree G1 and H1) to maintain stable gaits while tracking target velocities across rough surfaces. Both BPO and PPO were implemented using the RSL-RL framework, utilizing a large-scale parallelization of 4,096 environments per task.

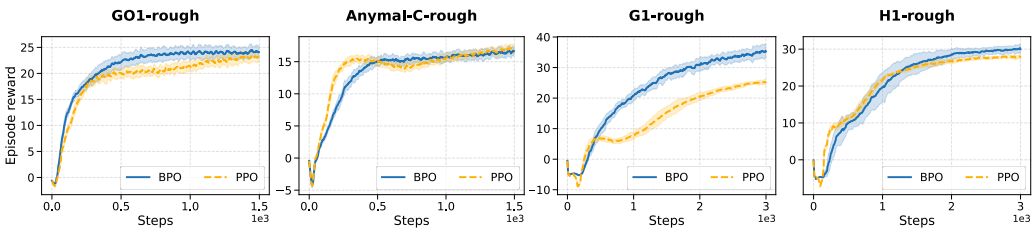

Figure 5: BPO versus PPO on IsaacLab environments. Shaded regions represent standard deviation across 5 random seeds. BPO substantially outperforms the baseline on challenging humanoid locomotion tasks while exhibiting more stable training dynamics.

The results in Figure 5 demonstrate that BPO is highly effective in complex robotic locomotion tasks. In particular, on G1-rough, BPO significantly outperforms the baseline to reach a higher performance ceiling. For the Go1-rough and H1-rough environment, BPO also slightly exceeds the final performance of PPO. Notably, across all four benchmarks, BPO exhibits enhanced training stability and smoother dynamics compared to the PPO baseline.

5.3 LLM Fine-Tuning with GBPO

We further evaluate GBPO against GRPO for large language model fine-tuning (Section 4.5). Specifically, we conduct experiments in the Test-Time Reinforcement Learning (TTRL, [35]) framework, fine-tuning the Qwen2.5-Math-1.5B model with GBPO and GRPO on the AIME-TTT and AMC-TTT benchmarks, and then compare their reasoning performance. The empirical results, illustrated in Figure 6, reveal that GBPO can maintain performance gains as the number of training epochs and clip ratio increase. Conversely, GRPO exhibits instability under these conditions. These findings highlight GBPO's potential as a more robust and stable alternative for the fine-tuning of large-scale models.

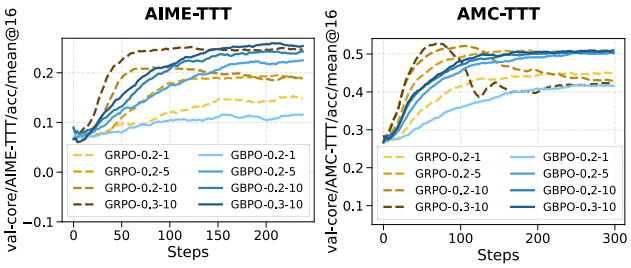

Figure 6: Performance of GRPO (green) and GBPO (blue) for fine-tuning Qwen2.5-Math-1.5B on AIME-TTT and AMC-TTT benchmarks. In the legend, the first and second numbers denote the clip ratio and the number of epochs, respectively.

5.4 Ratio Statistics Analysis

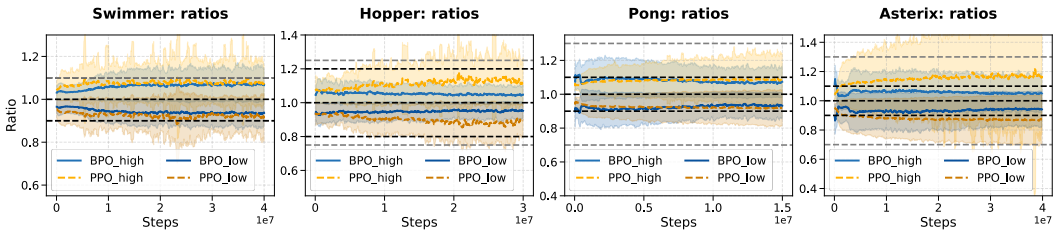

Figure 7: Analysis of ratio statistics. During the training process, we draw statistics of ratios $(\pi(a|s)/\pi_0(a|s))$ above and below 1.0 separately, corresponding to BPO/PPO_high and BPO/PPO_low. Solid lines and shaded regions represent the mean and standard deviations across 5 random seeds. Dashed black lines highlight 1.0 and clipped ranges for PPO; dashed gray lines show clipped ranges for BPO.

We analyze the statistics of importance weights (ratio $\pi(a|s)/\pi_0(a|s)$) during the training process. In MuJoCo environments (using the stable-baselines3 implementation), BPO maintains more stable ratio distributions than PPO, as illustrated in Figure 7. This difference in stability is more obvious in environments where BPO outperforms PPO (e.g., Hopper and Asterix). In IsaacLab environments (utilizing RSL-RL), learning rates are dynamically adjusted to maintain a target KL divergence. As shown in Figure 8, the adapted learning rates for PPO are often lower than those for BPO, suggesting more aggressive ratio updates that surpass the target KL divergence more frequently. The scales of the learning rates differ more in tasks where BPO shows a clear performance improvement (e.g., G1-rough). These findings suggest a strong correlation between the stability of ratio distributions and overall algorithmic performance. By effectively enforcing this stability, BPO allows for more stable performance improvement.

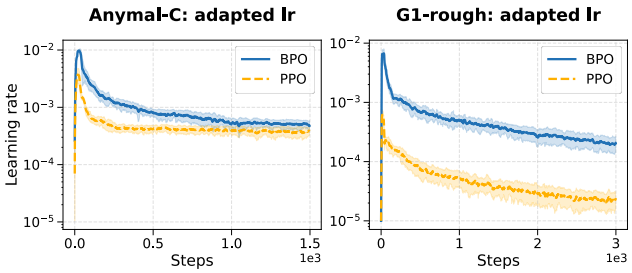

Figure 8: Adapted learning rates to match the target KL divergence in RSL-RL implementation.

5.5 Ablation Study

This section presents an ablation study of the impact of the value function, loss function, λ, and the coefficient of TV loss on the performance of the policy, within the G1-rough environment.

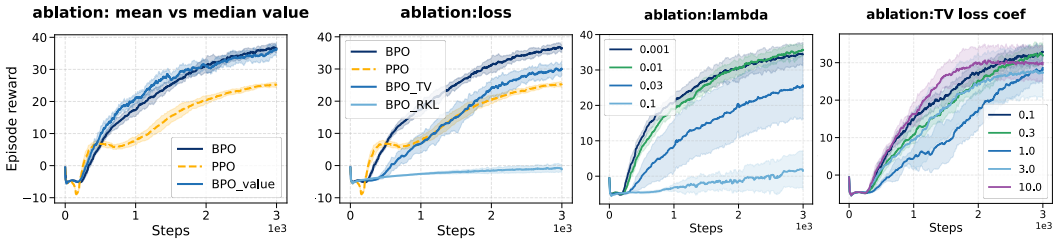

Figure 9: Ablation study of BPO components in G1-rough environment. Shaded regions represent standard deviation across 10 random seeds. From left to right: ablation of mean vs median value functions, loss functions, regularization weight λ and total variation (TV) weight α_1. Using the mean value instead of the median value yields similar final performance. The advantage-weighted total variation (ATV) loss provides better performance compared to TV and KL divergence. In general, smaller λ values lead to better performance, although making λ too small ($1e^{-4}$) slightly degrades performance. In practice, including the TV loss does not improve results.

Mean vs median value function. We evaluate the performance of the algorithm by substituting median advantages $\tilde{A}_{\pi_0}$ with the mean advantage A_{π_0}. As illustrated in the left panel of Figure 9, this simplification achieves performance comparable to the original BPO. This robustness likely stems from the low practical differences between median and mean values, caused by the specific return distribution and inherent value estimation errors. These results also suggest that this median-to-mean value simplification offers a compelling alternative when the computational overhead of learning the median value is high.

Divergence function ablation. As illustrated in the middle-left panel of Figure 9, the ATV loss consistently yields superior performance in both G1-rough and Anymal-C-rough environments. While the standard TV loss facilitates some learning, it fails to match the asymptotic performance of ATV. Conversely, KL divergence proves ineffective and fails to achieve successful policy convergence.

Sensitivity to λ. We conduct a hyperparameter sweep for λ in the G1-rough environment. As shown in the middle-right panel of Figure 9 , smaller values of λ generally lead to strong performance. Specifically, increasing λ from 10^{-3} to 10^{-2} may slightly improve asymptotic performance, but at

the cost of a reduced convergence rate. Conversely, excessively large values of λ prevent the learning process entirely.

Impact of TV loss regularization. We study the effect of the TV loss coefficient by incrementally increasing its weight relative to the ATV loss in the G1-rough environment. Although Corollary 4.5 suggests that both terms contribute to performance gains, the results in the right panel of Figure 9 indicate that explicitly adding a TV loss component does not improve performance in practice.

6 Conclusion

We introduced Bounded Ratio Reinforcement Learning (BRRL), a framework for policy optimization under bounded ratio constraints. We showed that the underlying optimization problem admits an analytic solution. Our main finding is that this optimal solution allows interpreting the PPO loss from a new perspective, connects to the cross-entropy method (CEM), and motivates a *theoretically grounded* variant, Bounded Policy Optimization (BPO). Empirically, BPO is consistently effective across a broad range of tasks, including robotic control and large-model fine-tuning. Despite the extensive evaluation with standard RL benchmarks, extending the experiments towards a broader range of LLM fine-tuning tasks remains a compelling future direction. Other future research directions include enhancing sample efficiency via advanced exploration, extending the framework to constrained MDPs, and adapting the algorithm for fine-tuning generative policies.

7 Acknowledgment

This work is in part supported by the Hasler Foundation ("Learn to learn safely" project, grant number: 21039), Swiss National Science Foundation under NCCR Automation, grant agreement 51NF40 180545, and the ETH AI Center.

References

[1] Abdullah Akgül, Gulcin Baykal, Manuel Haußmann, and Melih Kandemir. Overcoming non-stationary dynamics with evidential proximal policy optimization. *arXiv preprint arXiv:2503.01468*, 2025.

[2] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.

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

[4] Leif Doering, Daniel Schmidt, Moritz Melcher, Sebastian Kassing, Benedikt Wille, Tilman Aach, and Simon Weissmann. An approximate ascent approach to prove convergence of ppo. *arXiv preprint arXiv:2602.03386*, 2026.

[5] Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Implementation matters in deep policy gradients: A case study on ppo and trpo. *arXiv preprint arXiv:2005.12729*, 2020.

[6] Rasool Fakoor, Pratik Chaudhari, and Alexander J Smola. P3o: Policy-on policy-off policy optimization. In *Uncertainty in artificial intelligence*, pages 1017–1027. PMLR, 2020.

[7] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

[8] Taisuke Kobayashi. Proximal policy optimization with relative pearson divergence. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8416–8421. IEEE, 2021.

[9] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[10] L. D. Landau, E. M. Lifshitz, and L. P. Pitaevskii. *Statistical Physics: Theory of the Condensed State*, volume 9 of *Course of Theoretical Physics*. Butterworth-Heinemann, Oxford, 1980.

[11] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47):eabc5986, 2020.

[12] Boyi Liu, Qi Cai, Zhuoran Yang, and Zhaoran Wang. Neural trust region/proximal policy optimization attains globally optimal policy. *Advances in neural information processing systems*, 32, 2019.

[13] Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science robotics*, 7(62):eabk2822, 2022.

[14] Nikola Milosevic, Johannes Müller, and Nico Scherf. Central path proximal policy optimization. *arXiv preprint arXiv:2506.00700*, 2025.

[15] Mayank Mittal, Pascal Roth, James Tigue, Antoine Richard, Octi Zhang, Peter Du, Antonio Serrano-Muñoz, Xinjie Yao, René Zurbrügg, Nikita Rudin, et al. Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning. *arXiv preprint arXiv:2511.04831*, 2025.

[16] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

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

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

[19] Ilija Radosavovic, Tete Xiao, Bike Zhang, Trevor Darrell, Jitendra Malik, and Koushil Sreenath. Real-world humanoid locomotion with reinforcement learning. *Science Robotics*, 9(89):eadi9579, 2024.

[20] Antonin Raffin. Rl baselines3 zoo. `https://github.com/DLR-RM/rl-baselines3-zoo`, 2020.

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

[22] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

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

[24] Clemens Schwarke, Mayank Mittal, Nikita Rudin, David Hoeller, and Marco Hutter. Rsl-rl: A learning library for robotics research. *arXiv preprint arXiv:2509.10771*, 2025.

[25] Antonio Serrano-Munoz, Dimitrios Chrysostomou, Simon Bøgh, and Nestor Arana-Arexolaleiba. skrl: Modular and flexible library for reinforcement learning. *Journal of Machine Learning Research*, 24(254):1–9, 2023.

[26] 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.

[27] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.

[28] Charlie B Tan, Edan Toledo, Benjamin Ellis, Jakob N Foerster, and Ferenc Huszár. Beyond the boundaries of proximal policy optimization. *arXiv preprint arXiv:2411.00666*, 2024.

[29] Jiakang Wang, Runze Liu, Lei Lin, Wenping Hu, Xiu Li, Fuzheng Zhang, Guorui Zhou, and Kun Gai. Aspo: Asymmetric importance sampling policy optimization. *arXiv preprint arXiv:2510.06062*, 2025.

[30] Yuhui Wang, Hao He, and Xiaoyang Tan. Truly proximal policy optimization. In *Uncertainty in artificial intelligence*, pages 113–122. PMLR, 2020.

[31] Yuhui Wang, Hao He, Xiaoyang Tan, and Yaozhong Gan. Trust region-guided proximal policy optimization. *Advances in Neural Information Processing Systems*, 32, 2019.

[32] Zhiheng Xi, Xin Guo, Yang Nan, Enyu Zhou, Junrui Shen, Wenxiang Chen, Jiaqi Liu, Jixuan Huang, Zhihao Zhang, Honglin Guo, et al. Bapo: Stabilizing off-policy reinforcement learning for llms via balanced policy optimization with adaptive clipping. *arXiv preprint arXiv:2510.18927*, 2025.

[33] Zhengpeng Xie, Qiang Zhang, and Renjing Xu. Simple policy optimization. *arXiv preprint arXiv:2401.16025*, 2024.

[34] Deheng Ye, Zhao Liu, Mingfei Sun, Bei Shi, Peilin Zhao, Hao Wu, Hongsheng Yu, Shaojie Yang, Xipeng Wu, Qingwei Guo, et al. Mastering complex control in moba games with deep reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 6672–6679, 2020.

[35] Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, et al. Ttrl: Test-time reinforcement learning. *arXiv preprint arXiv:2504.16084*, 2025.

A Appendix

A.1 Proof of Theorem 4.1

Proof. Since $\forall\, s,\ \mathbb{E}_{a\sim\pi_0(\cdot|s)}[\rho] = 1$, we have from Equation (3)

$$\begin{aligned}
&L_{\pi_0}(\pi) = \eta(\pi_0) + \mathbb{E}_{s\sim d_{\pi_0}, a\sim\pi_0(\cdot|s)}[\rho A_{\pi_0}(s,a)] \\
=&\mathbb{E}_{s\sim d_{\pi_0}}[V_{\pi_0}(s)\cdot\mathbb{E}_{a\sim\pi_0(\cdot|s)}[\rho]] + \mathbb{E}_{s\sim d_{\pi_0}, a\sim\pi_0(\cdot|s)}[\rho A_{\pi_0}(s,a)] \\
=&\mathbb{E}_{s\sim d_{\pi_0}, a\sim\pi_0(\cdot|s)}[\rho V_{\pi_0}(s) + \rho A_{\pi_0}(s,a)] = \mathbb{E}_{s\sim d_{\pi_0}, a\sim\pi_0(\cdot|s)}[\rho Q_{\pi_0}(s,a)]
\end{aligned}$$

The original problem (9) can then be written as

$$\max_{\rho} \mathbb{E}_{\substack{s\sim d_{\pi_0}\\ a\sim\pi_0(\cdot|s)}}[\rho Q_{\pi_0}(s,a) - \lambda(\rho-1+\epsilon)\log(\rho-1+\epsilon) - \lambda(1+\epsilon-\rho)\log(1+\epsilon-\rho)],$$

with the normalization constraint

$$\forall\, s,\ \mathbb{E}_{a\sim\pi_0(\cdot|s)}[\rho] = 1.$$

This optimization problem can be decomposed into subproblems for each state s. Now, given a fixed state s, we solve the constrained optimization problem using the Lagrangian approach, with a Lagrangian multiplier denoted as $-\mu(s)$

$$\begin{aligned}
\mathcal{L}(\rho) :=&\mathbb{E}_{a\sim\pi_0(\cdot|s)}[\rho Q_{\pi_0}(s,a) - \lambda(\rho-1+\epsilon)\log(\rho-1+\epsilon) - \lambda(1+\epsilon-\rho)\log(1+\epsilon-\rho)] \\
&- \mu(s)(\mathbb{E}_{a\sim\pi_0(\cdot|s)}[\rho] - 1) \\
=:&\mathbb{E}_{a\sim\pi_0(\cdot|s)}[f(\rho)] + \mu(s),
\end{aligned}$$

where

$$f(\rho) := \rho Q_{\pi_0}(s,a) - \lambda(\rho-1+\epsilon)\log(\rho-1+\epsilon) - \lambda(1+\epsilon-\rho)\log(1+\epsilon-\rho) - \mu(s)\rho.$$

If $\mathcal{A}$ is continuous, we apply the calculus of variations

$$\begin{aligned}
&\frac{\partial}{\partial\rho} f(\rho) = 0 \\
\Rightarrow\quad &\frac{\partial}{\partial\rho}\left(\rho Q_{\pi_0}(s,a) - \lambda(\rho-1+\epsilon)\log(\rho-1+\epsilon) - \lambda(1+\epsilon-\rho)\log(1+\epsilon-\rho) - \mu(s)\rho\right) = 0 \\
\Rightarrow\quad &Q_{\pi_0}(s,a) - \lambda(\log(\rho-1+\epsilon) - \log(1+\epsilon-\rho)) - \mu(s) = 0 \\
\Rightarrow\quad &\log\frac{\rho-1+\epsilon}{1+\epsilon-\rho} = \frac{Q_{\pi_0}(s,a)-\mu(s)}{\lambda} \\
\Rightarrow\quad &\rho^* = 1 + \frac{\epsilon\exp\left(\frac{Q_{\pi_0}(s,a)-\mu(s)}{\lambda}\right) - \epsilon}{1+\exp\left(\frac{Q_{\pi_0}(s,a)-\mu(s)}{\lambda}\right)} = 1 + \epsilon\tanh\left(\frac{\tilde{A}_{\pi_0}}{2\lambda}\right)
\end{aligned} \tag{19}$$

The second derivative of the objective function can be computed as $-\frac{\lambda}{\rho-1+\epsilon} - \frac{\lambda}{1+\epsilon-\rho}$, which is negative for arbitrary $1-\epsilon < \rho < 1+\epsilon$. Therefore, ρ^* is the maximizer of the objective function.

The Lagrangian multiplier $\mu(s)$ should be chosen to satisfy the normalization constraint:

$$\mathbb{E}_{a\sim\pi_0(\cdot|s)}\left[1+\epsilon\tanh\left(\frac{\tilde{A}_{\pi_0}}{2\lambda}\right)\right] = 1 \quad\Leftrightarrow\quad \mathbb{E}_{a\sim\pi_0(\cdot|s)}\left[\tanh\left(\frac{\tilde{A}_{\pi_0}}{2\lambda}\right)\right] = 0.$$

Note that such $\mu(s)$ always exists because for all $\lambda > 0$, $\mathbb{E}_{a\sim\pi_0(\cdot|s)}\left[1+\epsilon\tanh\left(\frac{\tilde{A}_{\pi_0}}{2\lambda}\right)\right]$ is a smooth function of μ with the value range between $1-\epsilon$ and $1+\epsilon$. Now we show that the corresponding $\mu(s)$ is also the minimizer of $\mathbb{E}_{a\sim\pi_0(\cdot|s)}\left[g\left(\frac{Q_{\pi_0}(s,a)-\mu(s)}{\lambda}\right)\right]$, where $g := \ln(e^{-\frac{x}{2}} + e^{\frac{x}{2}})$:

$$\mathbb{E}_{\pi_0}\left[\frac{\partial g}{\partial u}\left(\frac{q-u}{\lambda}\right)\right] = 0 \ \Leftrightarrow\ \mathbb{E}_{\pi_0}\left[\frac{e^{\frac{q-u}{2\lambda}} - e^{-\frac{q-u}{2\lambda}}}{2(e^{-\frac{q-u}{2\lambda}} + e^{\frac{q-u}{2\lambda}})}\right] = 0 \ \Leftrightarrow\ \mathbb{E}_{\pi_0}\left[\tanh\left(\frac{q-u}{2\lambda}\right)\right] = 0,$$

where $\mathbb{E}_{\pi_0}$ abbreviates $\mathbb{E}_{a\sim\pi_0(\cdot|s)}$. Besides,

$$\frac{\partial^2 g}{\partial^2 u} = \frac{1}{4\lambda^2}\mathbb{E}_{a\sim\pi_0(\cdot|s)}\left[\frac{(e^{\frac{u-q}{2\lambda}}+e^{-\frac{u-q}{2\lambda}})^2-(e^{\frac{u-q}{2\lambda}}-e^{-\frac{u-q}{2\lambda}})^2}{2(e^{-\frac{u-q}{2\lambda}}+e^{\frac{u-q}{2\lambda}})^2}\right]\geq 0.$$

Therefore, the optimal $\mu(s)$ is the minimizer of g.

If $\mathcal{A}$ is discrete, for each fixed state s, we denote the vectorized $\rho(a)$, $Q_{\pi_0}(s,a)$, $H(\rho)$ and $\pi_0(a|s)$ as $\rho, Q, H, \pi \in \mathbb{R}^{|\mathcal{A}|}$. Then the Lagrangian $\mathcal{L}(\rho)$ can be expressed by

$$\mathcal{L}(\rho) := \pi^\top(\rho\odot Q - \lambda H - \mu\rho) + \mu,$$

where $\odot$ denotes elementwise product. Applying zero gradient w.r.t. ρ gives

$$(\text{diag}(Q) - \lambda\text{diag}(H') - \mu I)^\top\pi = \mathbf{0} \quad \Rightarrow \quad \pi(a|s)(Q_{\pi_0}(s,a) - \lambda H'(\rho) - \mu(s)) = 0,\ \forall\ a,$$

This gives the same expression as (19), therefore, the following proof steps are the same as the continuous case. □

A.2 Proof of Theorem 4.2

We start by proving the following Lemma:

Lemma A.1. *Define* $L^{\pi^*}_{\pi_0}(s) := \mathbb{E}_{a\sim\pi^*(\cdot|s)}[Q_{\pi_0}(s,a)]$, *consider* $\pi^*(a|s) = \left[1+\epsilon\tanh\left(\frac{\tilde{A}_{\pi_0}}{2\lambda}\right)\right]\pi_0(a|s)$ *from Theorem 4.1, we have*

$$L^{\pi^*}_{\pi_0}(s) = V_{\pi_0}(s) + \epsilon\mathbb{E}_{a\sim\pi_0(\cdot|s)}\left[\tanh\left(\frac{\tilde{A}_{\pi_0}(s,a)}{2\lambda}\right)\tilde{A}_{\pi_0}(s,a)\right].$$

Proof. We directly compute $L^{\pi^*}_{\pi_0}(s)$ as

$$\begin{aligned}
L^{\pi^*}_{\pi_0}(s) &= \mathbb{E}_{\pi_0}\left[\left(1+\epsilon\tanh\left(\frac{\tilde{A}_{\pi_0}(s,a)}{2\lambda}\right)\right)Q_{\pi_0}(s,a)\right]\\
&= \underbrace{\mathbb{E}_{\pi_0}[Q_{\pi_0}(s,a)]}_{V_{\pi_0}(s)} + \epsilon\mathbb{E}_{\pi_0}\left[\tanh\left(\frac{\tilde{A}_{\pi_0}(s,a)}{2\lambda}\right)Q_{\pi_0}(s,a)\right]\\
&= V_{\pi_0}(s) + \epsilon\mathbb{E}_{\pi_0}\left[\tanh\left(\frac{\tilde{A}_{\pi_0}(s,a)}{2\lambda}\right)\cdot(\mu(s)+\tilde{A}_{\pi_0}(s,a))\right]\\
&= V_{\pi_0}(s) + \epsilon\mathbb{E}_{\pi_0}\left[\tanh\left(\frac{\tilde{A}_{\pi_0}(s,a)}{2\lambda}\right)\tilde{A}_{\pi_0}(s,a)\right] + \epsilon\underbrace{\mathbb{E}_{\pi_0}\left[\tanh\left(\frac{\tilde{A}_{\pi_0}(s,a)}{2\lambda}\right)\right]}_{=0}\mu(s).
\end{aligned}$$

where we abbreviate $\mathbb{E}_{a\sim\pi_0(\cdot|s)}$ with $\mathbb{E}_{\pi_0}$. We have $\mathbb{E}_{\pi_0}\left[\tanh\left(\frac{\tilde{A}_{\pi_0}(s,a)}{2\lambda}\right)\right] = 0$ because of the normalization constraint for μ in Theorem 4.1. □

We now prove Theorem 4.2 using matrix representations for MDP with a discrete state and action space. The results for continuous spaces can be extended by using linear operators other than matrices.

Proof. (Theorem 4.2) We define $r_\pi \in \mathbb{R}^{|\mathcal{S}|}$ with $r_\pi(s) := \mathbb{E}_{a\sim\pi(\cdot|s),s'\sim P(s'|s,a)}[r(s,a,s')]$. The transition kernel $P_\pi \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ is defined with $P_\pi(s,s') := \sum_{a\sim\pi(\cdot|s)}\pi(a|s)p(s'|s,a)$. We denote $(I-\gamma P_\pi)^{-1} := (I+\gamma P_\pi+\gamma^2P_\pi^2+...)$. Then, given the initial distribution denoted as $d \in [0,1]^{|\mathcal{S}|}, d(s) := d_0(s)$, we can express the state visitation distribution $d_\pi \in [0,\frac{1}{1-\gamma}]^{|\mathcal{S}|}$ as $d_\pi^\top = d^\top(I-\gamma P_\pi)^{-1}$.

Let us define $V_\pi \in \mathbb{R}^{|\mathcal{S}|}$, where each component s corresponds to $V_\pi(s)$. The definition of the value function implies that

$$\begin{aligned} V_\pi &= r_\pi + \gamma P_\pi V_\pi \\ V_\pi &= (I - \gamma P_\pi)^{-1} r_\pi. \end{aligned} \tag{20}$$

We also have

$$\eta(\pi) = \mathbb{E}_{s\sim d_\pi, a\sim\pi(\cdot|s), s'\sim P(\cdot|s,a)}[r(s,a,s')] = d^\top (I - \gamma P_\pi)^{-1} r_\pi = d^\top V_\pi \tag{21}$$

We then define $L^{\pi_2}_{\pi_1} \in \mathbb{R}^{|\mathcal{S}|}$ with

$$L^{\pi_2}_{\pi_1} = r_{\pi_2} + \gamma P_{\pi_2} V_{\pi_1}, \tag{22}$$

which aligns with the definition of $L^{\pi^*}_{\pi_0}(s)$ in Lemma A.1. Let us denote $B_1 \in \mathbb{R}^{|\mathcal{S}|}$ with $B_1(s) := \mathbb{E}_{a\sim\pi_0(\cdot|s)}\left[\tanh\left(\frac{\tilde{A}_{\pi_0}(s,a)}{2\lambda}\right)\tilde{A}_{\pi_0}(s,a)\right]$. Then the state-wise result Lemma A.1 can be rewritten for the full state space as

$$L^{\pi^*}_{\pi_0} = V_{\pi_0} + \epsilon B_1, \tag{23}$$

where B_1 is positive along each of its components. Combining (22) and (23) gives

$$r_{\pi^*} + \gamma P_{\pi^*} V_{\pi_0} =: L^{\pi^*}_{\pi_0} = V_{\pi_0} + \epsilon B_1 \;\Rightarrow\; r_{\pi^*} = (I - \gamma P_{\pi^*}) V_{\pi_0} + \epsilon B_1 \tag{24}$$

On the other hand, applying (20) to π^* in combination with (24) gives

$$V_{\pi^*} = (I-\gamma P_{\pi^*})^{-1} r_{\pi^*} = (I-\gamma P_{\pi^*})^{-1}((I-\gamma P_{\pi^*})V_{\pi_0} + \epsilon B_1) = V_{\pi_0} + \epsilon (I-\gamma P_{\pi^*})^{-1} B_1. \tag{25}$$

Finally, we can obtain from (21)

$$\eta(\pi^*) := d^\top V_{\pi^*} = d^\top V_{\pi_0} + \epsilon d^\top (I - \gamma P_{\pi^*})^{-1} B_1 = \eta(\pi_0) + \epsilon \mathbb{E}_{s\sim d_{\pi^*}}[B_1(s)]$$

Applying the definition of $B_1(s)$ finishes the proof. □

A.3 Proof of Corollary 4.4

We start by proving the following Lemma on per-state performance improvement.

Lemma A.2. *Define* $L^{\pi_\theta}_{\pi_0}(s) := \mathbb{E}_{a\sim\pi_\theta(\cdot|s)}[Q_{\pi_0}(s,a)]$, $D^{ATV}_\theta(s) := \sum_a |(\pi_\theta(a|s) - \pi^*(a|s))A_{\pi_0}(s,a)|$, *we have*

$$L^{\pi_\theta}_{\pi_0}(s) \geq V_{\pi_0}(s) + \epsilon \mathbb{E}_{a\sim\pi_0(\cdot|s)}\left[\tanh\left(\frac{\tilde{A}_{\pi_0}(s,a)}{2\lambda}\right)\tilde{A}_{\pi_0}(s,a)\right] - D^{ATV}_\theta(s)$$

Proof. We first bound $|L^{\pi^*}_{\pi_0}(s) - L^{\pi_\theta}_{\pi_0}(s)|$ with $D^{ATV}_\theta(s)$

$$\begin{aligned} |L^{\pi^*}_{\pi_0}(s) - L^{\pi_\theta}_{\pi_0}(s)| =& |\mathbb{E}_{a\sim\pi^*(\cdot|s)}\left[Q_{\pi_0}(s,a)\right] - \mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[Q_{\pi_0}(s,a)\right]| \\ =& |\mathbb{E}_{a\sim\pi^*(\cdot|s)}\left[Q_{\pi_0}(s,a)\right] - V_{\pi_0}(s) - (\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[Q_{\pi_0}(s,a)\right] - V_{\pi_0}(s))| \\ =& |\mathbb{E}_{a\sim\pi^*(\cdot|s)}\left[A_{\pi_0}(s,a)\right] - \mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[A_{\pi_0}(s,a)\right]| \\ =& \left|\mathbb{E}_{a\sim\pi_0(\cdot|s)}\left[\left(\frac{\pi^*(a|s)}{\pi_0(a|s)} - \frac{\pi_\theta(a|s)}{\pi_0(a|s)}\right)A_{\pi_0}(s,a)\right]\right| \\ \leq& \mathbb{E}_{a\sim\pi_0(\cdot|s)}\left[\left|\left(\frac{\pi^*(a|s)}{\pi_0(a|s)} - \frac{\pi_\theta(a|s)}{\pi_0(a|s)}\right)A_{\pi_0}(s,a)\right|\right] \\ =& D^{ATV}_\theta(s) \end{aligned}$$

Then we have from Lemma A.1

$$L^{\pi_\theta}_{\pi_0}(s) \geq L^{\pi^*}_{\pi_0}(s) - D^P_\theta(s) = V_{\pi_0}(s) + \epsilon \mathbb{E}_{a\sim\pi_0(\cdot|s)}\left[\tanh\left(\frac{\tilde{A}_{\pi_0}(s,a)}{2\lambda}\right)\tilde{A}_{\pi_0}(s,a)\right] - D^{ATV}_\theta(s)$$

□

We now prove the Corollary 4.4.

Proof. Similar to the proof of 4.2, we start by defining $D_\theta^{ATV} \in \mathbb{R}^{|\mathcal{S}|}$ with each component as $D_\theta^{ATV}(s)$. Then we can express Lemma A.2 in the full state space as $L_{\pi_0}^{\pi_\theta} \succeq V_{\pi_0} + \epsilon B_1 - D_\theta^{ATV}$, where $\succeq$ denotes elementwise $\geq$. Following similar steps of the proof of Theorem 4.2, we obtain

$$r_{\pi_\theta} \succeq (I - \gamma P_{\pi_\theta}) V_{\pi_0} + \epsilon B_1 - D_\theta^{ATV}. \tag{26}$$

Multiplying both sides by $(I - \gamma P_{\pi_\theta})^{-1}$ from left gives

$$V_{\pi_\theta} \succeq V_{\pi_0} + (I - \gamma P_{\pi_\theta})^{-1}(\epsilon B_1 - D_\theta^{ATV}), \tag{27}$$

because all terms of $(I - \gamma P_{\pi_\theta})^{-1}$ are positive. Multiplying both sides by $d^\top$ finishes the proof. □

A.4 Proof of Corollary 4.5

Proof. From the inequality (27), we can obtain

$$\begin{aligned}
V_{\pi_\theta} \succeq & V_{\pi_0} + (I - \gamma P_{\pi_\theta})^{-1}(\epsilon B_1 - D_\theta^{ATV}) = V_{\pi_0} + \epsilon (I - \gamma P_{\pi_\theta})^{-1} B_1 - (I - \gamma P_{\pi_\theta})^{-1} D_\theta^{ATV} \\
= & V_{\pi_0} + \epsilon (I - \gamma P_{\pi_\theta})^{-1} B_1 - (I - \gamma P_{\pi_0})^{-1} D_\theta^{ATV} - ((I - \gamma P_{\pi_\theta})^{-1} - (I - \gamma P_{\pi_0})^{-1}) D_\theta^{ATV} \\
= & V_{\pi_0} + \epsilon (I - \gamma P_{\pi_\theta})^{-1} B_1 - (I - \gamma P_{\pi_0})^{-1} D_\theta^{ATV} \\
& \underbrace{- \gamma (I - \gamma P_{\pi_0})^{-1} \underbrace{(P_{\pi_\theta} - P_{\pi_0}) \underbrace{(I - \gamma P_{\pi_\theta})^{-1} D_\theta^{ATV}}_{:=X}}_{:=Y}}_{:=Z}
\end{aligned} \tag{28}$$

We can first bound X elementwise by

$$X \preceq (I - \gamma P_{\pi_\theta})^{-1} D_{\max}^{ATV} = \mathbf{1} \cdot \frac{D_{\max}^{ATV}}{1 - \gamma},$$

where $\preceq$ denotes elementwise smaller or equal to. Then we can bound each term Y by

$$\begin{aligned}
|Y(s)| = & |\sum_a (\pi_\theta(a|s) - \pi_0(a|s)) \sum_{s'} p(s'|s,a) X(s')| \\
\leq & \sum_a |\pi_\theta(a|s) - \pi_0(a|s)| \sum_{s'} p(s'|s,a) \max_{s'} |X(s')| \\
= & \sum_a |\pi_\theta(a|s) - \pi_0(a|s)| \max_{s'} |X(s')| \\
\leq & (\underbrace{\sum_a |\pi_0(a|s) - \pi^*(a|s)|}_{\mathbb{E}_{a \sim \pi_0(\cdot|s)}[|\frac{\pi^*(a|s)}{\pi_0(a|s)} - 1|]} + \sum_a |\pi_\theta(a|s) - \pi^*(a|s)|) \max_{s'} |X(s')| \\
\leq & (\epsilon + D_\theta^{TV}(s)) \cdot \frac{D_{\max}^{ATV}}{1 - \gamma}
\end{aligned}$$

Finally, Z satisfies

$$|Z| = |\gamma (I - \gamma P_{\pi_0})^{-1} Y| \preceq \gamma (I - \gamma P_{\pi_0})^{-1} (D_\theta^{TV} + \epsilon \cdot \mathbf{1}) \cdot \frac{D_{\max}^{ATV}}{1 - \gamma} \tag{29}$$

And

$$\begin{aligned}
|d^\top Z| & \leq \gamma d^\top (I - \gamma P_{\pi_0})^{-1} (D_\theta^{TV} + \epsilon \cdot \mathbf{1}) \cdot \frac{D_{\max}^{ATV}}{1 - \gamma} \\
& = \frac{\gamma D_{\max}^{ATV}}{1 - \gamma} \mathbb{E}_{s \sim d_{\pi_0}}[D_\theta^{TV}(s)] + \frac{\gamma \epsilon D_{\max}^{ATV}}{(1 - \gamma)^2}
\end{aligned} \tag{30}$$

Now let us consider the second term of (28) as

$$\epsilon(I-\gamma P_{\pi_\theta})^{-1}B_1 = \epsilon(I-\gamma P_{\pi^*})^{-1}B_1 - \epsilon((I-\gamma P_{\pi^*})^{-1} - (I-\gamma P_{\pi_\theta})^{-1})B_1.$$

With similar techniques for bounding X, Y and Z from (28) to (29), we have

$$\begin{aligned}
&|d^\top((I-\gamma P_{\pi^*})^{-1} - (I-\gamma P_{\pi_\theta})^{-1})B_1| \le \gamma d^\top (1-\gamma P_{\pi^*})^{-1} D_\theta^{TV} \cdot \frac{\tilde{\delta}}{1-\gamma} \\
&\le \gamma d^\top (1-\gamma P_{\pi^*})^{-1}\mathbf{1} \cdot D_{\max}^{TV} \cdot \frac{\tilde{\delta}}{1-\gamma} = \frac{\gamma\tilde{\delta} D_{\max}^{TV}}{(1-\gamma)^2}
\end{aligned}$$

Combining this with (30) and (28) finishes the proof. □

The results for continuous spaces can be extended by using linear operators other than matrices.

A.5 Proof of Proposition 4.6

We use A_0 to abbreviate A_{π_0}. When $A_0 > 0$, the negative PPO objective for a fixed state-action pair can be further expressed as

$$\begin{aligned}
l^{PPO}(\rho) &= \begin{cases} -\rho A_0, & \rho \le 1+\epsilon \\ -(1+\epsilon)A_0, & \rho > 1+\epsilon \end{cases} \\
\Rightarrow \quad l'(\rho) := l^{PPO}(\rho) + (1+\epsilon)A_0 &= \begin{cases} [(1+\epsilon)-\rho]A_0, & \rho \le 1+\epsilon \\ 0, & \rho > 1+\epsilon \end{cases},
\end{aligned}$$

where l' is constructed from adding a constant $(1+\epsilon)A_0$ to l^{PPO}. Similarly, we construct l' for $A_0 \le 0$:

$$\begin{aligned}
l^{PPO}(\rho) &= \begin{cases} -\rho A_0, & \rho \ge 1-\epsilon \\ -(1-\epsilon)A_0, & \rho < 1-\epsilon \end{cases} \\
\Rightarrow \quad l'(\rho) := l^{PPO}(\rho) + (1-\epsilon)A_0 &= \begin{cases} [(1-\epsilon)-\rho]A_0, & \rho \ge 1-\epsilon \\ 0, & \rho < 1-\epsilon \end{cases},
\end{aligned}$$

We can rearrange l' as

$$l'(\rho) = \begin{cases} |A_0| \cdot |\rho - (1+\text{sign}(A_0)\cdot\epsilon)|, & |\rho-1| \le \epsilon \\ 0, & |\rho-1| \ge \epsilon \end{cases}$$

Then we have

$$\begin{aligned}
\mathbb{E}_{s\sim d_{\pi_0}, a\sim\pi_0(\cdot|s)}[l'(\rho)] &= \mathbb{E}_{s\sim d_{\pi_0}, a\sim\pi_0(\cdot|s)}[l^{PPO}(\rho)] \\
&\quad + \mathbb{E}_{s\sim d_{\pi_0}, a\sim\pi_0(\cdot|s)}[(1+\epsilon)\mathbb{I}(A_0>0)A_0 + (1-\epsilon)\mathbb{I}(A_0\le 0)A_0],
\end{aligned}$$

where the second term is a constant independent of ρ. Therefore, l' is an equivalent loss function to l^{PPO} for solving the optimal ρ.

A.6 Proof of Corollary 4.7

Similar to Section A.1, for each fixed s, the Lagrangian of problem (18) can be written as

$$\begin{aligned}
\mathcal{L}(\rho) =& \mathbb{E}_{a\sim\pi_0(\cdot|s)}\left[\rho Q_{\pi_0}(s,a) - \lambda(\rho-c_l)\log(\rho-c_l) - \lambda(c_h-\rho)\log(c_h-\rho) - \lambda\log\frac{c_h-1}{1-c_l}\rho\right] \\
&-\mu(s)(\mathbb{E}_{a\sim\pi_0(\cdot|s)}[\rho]-1)
\end{aligned}$$

For continuous MDP, applying $\frac{\partial L}{\partial \rho} = 0$ gives

$$
\begin{aligned}
& Q_{\pi_0}(s,a) - \lambda \log(\rho - c_l) + \lambda \log(c_h - \rho) - \lambda \log \frac{c_h - 1}{1 - c_l} - \mu(s) = 0 \\
\Leftrightarrow \quad & Q_{\pi_0}(s,a) - \mu(s) = \lambda \left(\log \frac{\rho - c_l}{c_h - \rho} + \log \frac{c_h - 1}{1 - c_l} \right) \\
\Leftrightarrow \quad & \frac{\rho - c_l}{c_h - \rho} = \frac{1 - c_l}{c_h - 1} \exp \frac{Q_{\pi_0}(s,a) - \mu(s)}{\lambda} \\
\Leftrightarrow \quad & \rho = \frac{c_l + c_h \cdot \frac{1-c_l}{c_h - 1} \exp \frac{Q_{\pi_0}(s,a) - \mu(s)}{\lambda}}{1 + \frac{1-c_l}{c_h - 1} \exp \frac{Q_{\pi_0}(s,a) - \mu(s)}{\lambda}} = c_l + \frac{c_h - c_l}{1 + \frac{c_h - 1}{1 - c_l} \exp \frac{\mu(s) - Q_{\pi_0}(s,a)}{\lambda}}
\end{aligned}
$$

Now we derive the conditions that need to be satisfied by $\mu'_{\pi_0}(s)$. Specifically, having ρ normalized gives

$$
\begin{aligned}
& \mathbb{E}_{a \sim \pi_0(\cdot|s)} \left[c_l + \frac{c_h - c_l}{1 + \frac{c_h - 1}{1 - c_l} \exp \frac{\mu(s) - Q_{\pi_0}(s,a)}{\lambda}} \right] = 1 \\
\Leftrightarrow \, & \mathbb{E}_{a \sim \pi_0(\cdot|s)} \left[\frac{c_h - c_l}{1 + \frac{c_h - 1}{1 - c_l} \exp \frac{\mu(s) - Q_{\pi_0}(s,a)}{\lambda}} \right] = 1 - c_l \\
\Leftrightarrow \, & \mathbb{E}_{a \sim \pi_0(\cdot|s)} \left[\frac{c_h - c_l - (1 - c_l) - (c_h - 1) \exp \frac{\mu(s) - Q_{\pi_0}(s,a)}{\lambda}}{1 + \frac{c_h - 1}{1 - c_l} \exp \frac{\mu(s) - Q_{\pi_0}(s,a)}{\lambda}} \right] = 0 \\
\Leftrightarrow \, & \mathbb{E}_{a \sim \pi_0(\cdot|s)} \left[\frac{1 - \exp \frac{\mu(s) - Q_{\pi_0}(s,a)}{\lambda}}{1 + \frac{c_h - 1}{1 - c_l} \exp \frac{\mu(s) - Q_{\pi_0}(s,a)}{\lambda}} \right] = 0
\end{aligned}
$$

On the other hand, $\frac{\partial g'}{\partial x} = 0$ from g' defined in Corollary 4.7 gives

$$
\begin{aligned}
& \mathbb{E}_{a \sim \pi_0(\cdot|s)} \left[\frac{\frac{c_h - 1}{c_h - c_l} e^{\frac{c_h - 1}{c_h - c_l} x} - \frac{c_h - 1}{1 - c_l} \cdot \frac{1 - c_l}{c_h - c_l} e^{-\frac{1 - c_l}{c_h - c_l} x}}{e^{\frac{c_h - 1}{c_h - c_l} x} + \frac{c_h - 1}{1 - c_l} e^{-\frac{1 - c_l}{c_h - c_l} x}} \right] = 0 \\
\Leftrightarrow \quad & \mathbb{E}_{a \sim \pi_0(\cdot|s)} \left[\frac{e^{\frac{c_h - 1}{c_h - c_l} x} - e^{-\frac{1 - c_l}{c_h - c_l} x}}{e^{\frac{c_h - 1}{c_h - c_l} x} + \frac{c_h - 1}{1 - c_l} e^{-\frac{1 - c_l}{c_h - c_l} x}} \right] = 0 \\
\Leftrightarrow \quad & \mathbb{E}_{a \sim \pi_0(\cdot|s)} \left[\frac{1 - e^{-x}}{1 + \frac{c_h - 1}{1 - c_l} e^{-x}} \right] = 0,
\end{aligned}
$$

where assigning $x = \frac{\tilde{A}'_{\pi_0}}{\lambda}$ recovers the normalization condition above. For a discrete MDP, similar to the proof of Theorem 4.2, we can obtain the same Lagrangian as the continuous case.

A.7 Monotonic Guarantees for Asymmetric Bounded Ratio RL

Corollary A.3 (Asymmetric monotonic performance guarantee)**.** *The optimal policies in Theorem 4.7 satisfy*

$$
\begin{aligned}
\eta(\pi^*) &\geq \eta(\pi_0) + (c_h - 1) \mathbb{E}_{s \sim d_{\pi^*}, a \sim \pi_0(\cdot|s)} \left[\frac{1 + e^{-\tilde{A}'_{\pi_0}/\lambda}}{1 + \frac{c_h - 1}{1 - c_l} e^{-\tilde{A}'_{\pi_0}/\lambda}} \cdot \tanh \left(\frac{\tilde{A}_{\pi_0}}{2\lambda} \right) \tilde{A}_{\pi_0} \right] \\
&=: \eta(\pi_0) + (c_h - 1) B',
\end{aligned}
$$

where $\tilde{A}_{\pi_0}$ abbreviates $\tilde{A}_{\pi_0}(s,a)$, B' is a non-negative constant given fixed π_0.

Proof. We start by deriving $L^{\pi^*}_{\pi_0}(s)$, similar to Lemma A.1.

$$\begin{aligned}
L^{\pi^*}_{\pi_0}(s) =& \mathbb{E}_{a\sim\pi_0(\cdot|s)}\left[\frac{\pi^*(a|s)}{\pi_0(a|s)}Q_{\pi_0}(s,a)\right] \\
=& V_{\pi_0}(s) + \mathbb{E}_{a\sim\pi_0(\cdot|s)}\left[\left(\frac{\pi^*(a|s)}{\pi_0(a|s)}-1\right)Q_{\pi_0}(s,a)\right] \\
=& V_{\pi_0}(s) + \mu_{\pi_0}(s)\cdot\underbrace{\mathbb{E}_{a\sim\pi_0(\cdot|s)}\left[\left(\frac{\pi^*(a|s)}{\pi_0(a|s)}-1\right)\right]}_{=0} \\
&+ \mathbb{E}_{a\sim\pi_0(\cdot|s)}\left[\left(\frac{\pi^*(a|s)}{\pi_0(a|s)}-1\right)\tilde{A}_{\pi_0}(s,a)\right]
\end{aligned} \tag{31}$$

where $\mathbb{E}_{a\sim\pi_0(\cdot|s)}\left[\left(\frac{\pi^*(a|s)}{\pi_0(a|s)}-1\right)\right]=0$ is because of the normalization constraints enforced by the definition of $\mu'_{\pi_0}(s)$. We now compute $\frac{\pi^*(a|s)}{\pi_0(a|s)}-1$ from Corollary 4.7:

$$\begin{aligned}
\frac{\pi^*(a|s)}{\pi_0(a|s)}-1 =& c_l + \frac{c_h-c_l}{1+\frac{c_h-1}{1-c_l}\exp(-\tilde{A}'_{\pi_0}/\lambda)} - 1 \\
=& \frac{c_h-c_l+c_l-1-(c_h-1)\exp(-\tilde{A}'_{\pi_0}/\lambda)}{1+\frac{c_h-1}{1-c_l}\exp(-\tilde{A}'_{\pi_0}/\lambda)} \\
=& (c_h-1)\cdot\frac{1-\exp(-\tilde{A}'_{\pi_0}/\lambda)}{1+\frac{c_h-1}{1-c_l}\exp(-\tilde{A}'_{\pi_0}/\lambda)} \\
=& (c_h-1)\cdot\frac{1+\exp(-\tilde{A}'_{\pi_0}/\lambda)}{1+\frac{c_h-1}{1-c_l}\exp(-\tilde{A}'_{\pi_0}/\lambda)}\cdot\frac{1-\exp(-\tilde{A}'_{\pi_0}/\lambda)}{1+\exp(-\tilde{A}'_{\pi_0}/\lambda)} \\
=& (c_h-1)\cdot\frac{1+\exp(-\tilde{A}'_{\pi_0}/\lambda)}{1+\frac{c_h-1}{1-c_l}\exp(-\tilde{A}'_{\pi_0}/\lambda)}\cdot\tanh\left(\frac{\tilde{A}'_{\pi_0}}{2\lambda}\right)
\end{aligned}$$

Apply this into (31) gives

$$L^{\pi^*}_{\pi_0}(s) = V_{\pi_0}(s) + (c_h-1)\cdot\mathbb{E}_{a\sim\pi_0(\cdot|s)}\left[\frac{1+\exp(-\tilde{A}'_{\pi_0}/\lambda)}{1+\frac{c_h-1}{1-c_l}\exp(-\tilde{A}'_{\pi_0}/\lambda)}\cdot\tanh\left(\frac{\tilde{A}'_{\pi_0}}{2\lambda}\right)\tilde{A}'_{\pi_0}\right]$$

Following the same steps of proof of Theorem 4.2 in Appendix A.2, we can apply expectations over states and finish the proof. Notably, since $\frac{1+e^{-\tilde{A}'_{\pi_0}/\lambda}}{1+\frac{c_h-1}{1-c_l}e^{-\tilde{A}'_{\pi_0}/\lambda}}$ is always positive, $\frac{\pi^*(a|s)}{\pi_0(a|s)}-1$ still has the same sign as $\tilde{A}'_{\pi_0}$, which makes B' non-negative. □

With Corollary A.3, one can also derive corresponding corollaries for asymmetric BPO like Corollary 4.4 and 4.5.

A.8 Hyperparameters

The hyperparameters for BPO and PPO in RL environments are summarized in Table 2 and Table 3. For the GBPO implementation, we set the discount factor $\gamma = 1$ and utilize a mini-batch size of 1. Other hyperparameters are the same as those in the original training scripts provided by TTRL [35]. All the experiments are conducted on 4 x NVIDIA H100 GPUs. We set the group size to 32 and the maximum sequence length to 4,096 tokens.

Empirical Tuning Observations: While BPO can generally be initialized using hyperparameters tuned for PPO, specific adjustments often yield superior performance. Empirically, we found that increasing the clip ratio by 0.1 and doubling the number of epochs (e.g., from 5 to 10) can sometimes enhance stability and results. Furthermore, BPO exhibits a higher sensitivity to the entropy coefficient; in many environments, an entropy weight 10^{-1} smaller than the optimal PPO setting is sufficient to maintain adequate exploration without destabilizing the policy.

Table 2: Hyperparameters of BPO for benchmarking environments. We take $\lambda = 0.001$, $\alpha_1 = 0$ and $w_1 = w_2 = 0.5$ across all environments. The GAE-λ is set to 0.95 for all environments except 0.98 for Swimmer. ADP abbreviates adaptive learning rates based on the KL-divergence, according to [24].

Envs	batch size	clip	ent_coef	gamma	lr	n_epochs	n_steps	n_envs
Atari	256	0.3	0.001	0.98	$2.5e^{-4}$	5	128	8
Ant-v4	256	0.3	0	0.99	$1e^{-4}$	10	2048	1
Humanoid-v4	128	0.2	0	0.99	$1e^{-4}$	5	512	1
Hopper-v4	32	0.25	0	0.999	$9.808e^{-5}$	10	512	4
Swimmer-v4	256	0.1	0	0.9999	$3e^{-4}$	10	1024	4
Go1-Rough	24576	0.3	0.001	0.99	ADP	10	24	4096
Anymal-C	24576	0.25	0	0.99	ADP	5	24	4096
G1-Rough	24576	0.2	0	0.99	ADP	10	24	4096
H1-Rough	24576	0.2	0	0.99	ADP	5	24	4096

Table 3: Hyperparameters of PPO for benchmarking environments, based on RL-Zoo [20]. The GAE-λ is set to 0.95 for all environments except 0.98 for Swimmer, 0.99 for Hopper, 0.9 for Humanoid, and 0.8 for Ant. ADP abbreviates adaptive learning rates based on the KL-divergence, according to [24].

Envs	batch size	clip	ent_coef	gamma	lr	n_epochs	n_steps	n_envs
Atari	256	0.2	0.01	0.98	$2.5e^{-4}$	4	128	8
Ant-v4	32	0.1	$4.96e^{-7}$	0.98	$1.9e^{-5}$	10	512	1
Humanoid-v4	256	0.3	0.00238	0.98	$3.57e^{-5}$	5	512	1
Hopper-v4	32	0.25	0	0.999	$9.808e^{-5}$	10	512	4
Swimmer-v4	256	0.1	0	0.9999	$6e^{-4}$	10	1024	4
Go1-Rough	24576	0.2	0.01	0.99	ADP	5	24	4096
Anymal-C	24576	0.2	0.005	0.99	ADP	5	24	4096
G1-Rough	24576	0.2	0.008	0.99	ADP	5	24	4096
H1-Rough	24576	0.2	0.01	0.99	ADP	5	24	4096