Dual Approximation Policy Optimization

Zhihan Xiong
University of Washington
Seattle, WA 98195, USA
zhihanx@cs.washington.edu

Maryam Fazel
University of Washington
Seattle, WA 98195, USA
mfazel@uw.edu

Lin Xiao
FAIR at Meta
Seattle, WA 98109, USA
linx@meta.com

Abstract

We propose Dual Approximation Policy Optimization (DAPO), a framework that incorporates general function approximation into policy mirror descent methods. In contrast to the popular approach of using the L_2-norm to measure function approximation errors, DAPO uses the dual Bregman divergence induced by the mirror map for policy projection. This duality framework has both theoretical and practical implications: not only does it achieve fast linear convergence with general function approximation, but it also includes several well-known practical methods as special cases, immediately providing strong convergence guarantees.

1 Introduction

Policy gradient methods represent a paradigm shift in reinforcement learning from value-based methods [Watkins, 1989, Puterman, 1994, Bertsekas, 2015] to a more direct approach of policy optimization [Williams, 1992, Sutton et al., 1999, Konda and Tsitsiklis, 1999]. In particular, the natural policy gradient (NPG) method of Kakade [2001] inspired later development of trust region policy optimization (TRPO) [Schulman et al., 2015] and proximal policy optimization (PPO) Schulman et al. [2017], both with great empirical success.

These successes ignited considerable efforts to understand policy gradient methods from a theoretical perspective. Among them, Neu et al. [2017] first connected NPG with the mirror descent (MD) algorithm [Nemirovski and Yudin, 1983, Beck and Teboulle, 2003], which led to a more general class of policy mirror descent (PMD) methods. Convergence guarantees for tabular PMD methods progressed from sublinear convergence [Shani et al., 2020a, Agarwal et al., 2021] to linear convergence [Xiao, 2022, Lan, 2023, Johnson et al., 2023]. Then the linear convergence results were extended to PMD methods with linear function approximation [Yuan et al., 2022], and more recently with general function approximation [Alfano et al., 2023].

However, the progresses of PMD on the empirical and theoretical fronts are more or less disjoint, especially concerning general function approximation. One one hand, Tomar et al. [2020] and Vaswani et al. [2021] derived practical algorithms from the MD principle, but with no or limited convergence guarantees. On the other hand, Alfano et al. [2023] proposed Approximate Mirror Policy Optimization (AMPO), a PMD framework that has linear convergence guarantee with general function approximation, but has limited empirical success (see our empirical study in Section 5).

In this paper, we aim to bridge this gap between theory and practice by proposing Dual Approximation Policy Optimization (DAPO), a new PMD framework that incorporates general function approximation. In contrast to AMPO, which uses the squared L_2-norm to measure the function approximation error and tries to minimize it for policy update, DAPO uses the *dual Bregman divergence* generated by the mirror map used for policy projection.

We present several instantiations of DAPO using different mirror maps and prove linear convergence rates for two variants, DAPO-L_2 equipped with the squared L_2-norm as mirror map, and DAPO-KL with the negative entropy. We show that DAPO-KL includes two state-of-the-art practical

38th Workshop on Aligning Reinforcement Learning Experimentalists and Theorists (ARLET 2024).

algorithms as special cases: Soft Actor-Critic (SAC) of Haarnoja et al. [2018a] and Mirror Descent Policy Optimization (MDPO) of Tomar et al. [2020], thus immediately providing them with strong convergence guarantees. We compare DAPO with SAC and AMPO on several standard MuJoCo benchmark tasks to demonstrate the effectiveness of this duality framework.

In addition, in order to work with negative entropy restricted on the simplex in the setting of general function approximation, we extend the MD theory to work with mirror maps whose gradient mapping and conjugate mapping are not inverses of each other, a technical contribution of independent interest.

2 Preliminaries

We first review the background of Markov decision processes (MDPs) and the general MD algorithm.

2.1 Markov Decision Processes

Let $\Delta(\mathcal{X}) = \{p \in \mathbb{R}^{|\mathcal{X}|} \mid \sum_{x \in \mathcal{X}} p_x = 1 \text{ and } p_x \geq 0, \forall x\}$ denote the probability simplex over an arbitrary finite set $\mathcal{X}$. We consider an infinite-horizon Markov Decision Process (MDP), denoted as $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, c, \gamma)$, where $\mathcal{S}$ is a finite state space, $\mathcal{A}$ is a finite action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \mapsto \Delta(\mathcal{S})$ is the transition kernel, $c : \mathcal{S} \times \mathcal{A} \mapsto [0, 1]$ is the single-step cost function and $\gamma \in (0, 1)$ is the discount factor. A stationary policy is defined as a function $\pi : \mathcal{S} \mapsto \Delta(\mathcal{A})$ such that π_s is a probability distribution over $\mathcal{A}$ for each $s \in \mathcal{S}$. At each time t, an agent with policy π takes an action $a_t \sim \pi_{s_t}$, which sends the MDP to the new state $s_{t+1} \sim \mathcal{P}(s_t, a_t)$ and incurs a single-step cost $c(s_t, a_t)$.

Our main objective is to find a policy that minimizes the accumulated, discounted cost starting from an initial state distribution $\rho \in \Delta(\mathcal{S})$. Formally, it is defined as $V_\rho^\pi = \mathbb{E}_{s \sim \rho}[V_s^\pi]$, where

$$V_s^\pi = \mathbb{E}_{a_t \sim \pi_{s_t}}\left[\sum_{t=0}^\infty \gamma^t c(s_t, a_t) \mid s_0 = s\right]. \tag{1}$$

The corresponding Q-value function under policy π and state-action pair (s, a) is defined as

$$Q_{s,a}^\pi = \mathbb{E}_{a_t \sim \pi_{s_t}}\left[\sum_{t=0}^\infty \gamma^t c(s_t, a_t) \mid s_0 = s, a_0 = a\right]. \tag{2}$$

We use $Q_s^\pi \in \mathbb{R}^{|\mathcal{A}|}$ to denote the vector $\left[Q_{s,a}^\pi\right]_{a \in \mathcal{A}}$ and we immediately have $V_s^\pi = \langle Q_s^\pi, \pi_s \rangle$.

With initial distribution $\rho \in \Delta(\mathcal{S})$, we define the discounted state-visitation distribution under π as

$$d_{\rho,s}^\pi = (1 - \gamma) \sum_{t=0}^\infty \gamma^t \, \mathbb{P}_{s_0 \sim \rho}^\pi (s_t = s), \tag{3}$$

where $\mathbb{P}_{s_0 \sim \rho}^\pi (s_t = s)$ represents the probability that $s_t = s$ if the agent follows policy π and the initial state s_0 is sampled from distribution ρ. We can easily verify that $\sum_{s \in \mathcal{S}} d_{\rho,s}^\pi = 1$ and thus $d_\rho^\pi \in \mathbb{R}^{|\mathcal{S}|}$ is a valid probability distribution. Meanwhile, by truncating all terms with $t \geq 1$ in the sum in (3), we obtain $d_{\rho,s}^\pi \geq (1 - \gamma)\rho_s$ for any $s \in \mathcal{S}$.

The gradient of V_ρ^π with respect to π is given by the policy gradient theorem [Sutton et al., 1999] as

$$\nabla_s V_\rho^\pi := \frac{\partial V_\rho^\pi}{\partial \pi_s} = \frac{1}{1-\gamma} d_{\rho,s}^\pi Q_s^\pi \in \mathbb{R}^{|\mathcal{A}|}. \tag{4}$$

Then, we define $\nabla V_\rho^\pi \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{A}|}$ as the concatenation of $\nabla_s V_\rho^\pi$ for all $s \in \mathcal{S}$.

2.2 Mirror Descent

Mirror descent (MD) is a general framework for the construction and analysis of optimization algorithms [Nemirovski and Yudin, 1983]. Its key machinery is a pair of conjugate mirror maps that map the iterates of an optimization algorithm back-and-forth between a primal space and a dual space. We follow the common practice of defining the mirror maps with the gradient mapping of a convex function of *Legendre-type* [Rockafellar, 1970, Section 26].

Let's first define *Bregman divergence* and *Bregman projection*. Suppose that Ψ is a convex function of Legendre type. It induces a Bregman divergence between any $x \in \operatorname{dom} \Phi$ and $y \in \operatorname{int}(\operatorname{dom} \Phi)$:

$$D_\Phi(x, y) = \Phi(x) - \Phi(y) - \langle \nabla \Phi(y), x - y \rangle. \tag{5}$$

Let $\mathcal{C} \in \operatorname{dom} \Phi$ be a closed convex set. The *Bregman projection* of any $y \in \operatorname{int}(\operatorname{dom} \Phi)$ onto $\mathcal{C}$ is

$$\operatorname{proj}_{\mathcal{C}}^\Phi(y) = \arg\min_{x \in \mathcal{C}} D_\Phi(x, y). \tag{6}$$

Properties of Bregman projection can be found in, e.g., Bauschke and Borwein [1997].

Now consider the problem of minimizing a convex function $f : \mathcal{X} \to \mathbb{R} \cup \{\infty\}$ over a closed convex set $\mathcal{C} \subset \operatorname{dom} \Phi$. We use the presentation of MD given by Bubeck [2015]: at each iteration k,

1. Given $x^{(k)}$, find $y^{(k+1)}$ such that
$$\nabla \Phi(y^{(k+1)}) = \nabla \Phi(x^{(k)}) - \eta_k g^{(k)}. \tag{7}$$
where η_k is the step size and $g^{(k)}$ is the gradient $\nabla f(x^{(k)})$ or a sub-gradient of f at $x^{(k)}$.
2. Compute $x^{(k+1)} = \operatorname{proj}_{\mathcal{C}}^{\Phi}(y^{(k+1)})$.

Define the *conjugate function* of Φ as $\Phi^*(x^*) = \sup_{x \in \operatorname{dom} \Phi} \{\langle x, x^* \rangle - \Phi(x)\}$. Then, using the definition in (6) and the identity $\nabla \Phi^*(\nabla \Phi(x)) = x$, we can express it more compactly as
$$x^{(k+1)} = \underset{x \in \mathcal{C}}{\arg\min}\, D_\Phi\Big(x, \nabla \Phi^*\big(\nabla \Phi(x^{(k)}) - \eta_k g^{(k)}\big)\Big), \tag{8}$$
which can be further simplified to [Beck and Teboulle, 2003, Bubeck et al., 2012]
$$x^{(k+1)} = \underset{x \in \mathcal{C}}{\arg\min} \left\{ \eta_k \langle g^{(k)}, x \rangle + D_\Phi(x, x^{(k)}) \right\}. \tag{9}$$

Next we discuss three examples of the MD algorithm for solving $\min_{x \in \Delta} f(x)$, where Δ is the simplex. Each leads to a variant of the DAPO method we will present in Section 3.

Example 2.1 (Squared L_2-norm)**.** Let $\Phi(x) = \frac{1}{2}\|x\|_2^2$, which is Legendre type with $\operatorname{int}(\operatorname{dom} \Phi) = \operatorname{dom} \Phi = \mathbb{R}^n$. We have $\Phi^*(x^*) = \frac{1}{2}\|x^*\|_2^2$, $\nabla \Phi(x) = x$, $\nabla \Phi^*(x^*) = x^*$, and $\mathcal{D}_\Phi(x, y) = \frac{1}{2}\|x - y\|_2^2$. In this case, the MD algorithm (8) becomes the classical projected gradient method
$$x^{(k+1)} = \underset{x \in \Delta}{\arg\min} \left\| x - \left(x^{(k)} - \eta_k \nabla f(x^{(k)})\right) \right\|_2^2.$$

Example 2.2 (Negative entropy on $\mathbb{R}_+^n$)**.** Consider the negative entropy $\Phi(x) = \sum_i (x_i \log(x_i) - x_i)$ with $\operatorname{dom} \Phi = \mathbb{R}_+^n$ (and the convention $0 \log 0 = 0$). It is of Legendre type, with $\Phi^*(x^*) = \sum_i \exp(x_i^*)$, $\nabla \Phi(x) = \log(x)$ and $\nabla \Phi^*(x^*) = \exp(x^*)$, where $\log$ and $\exp$ apply component-wise to vectors. For any $x \in \mathbb{R}_+^n$ and $y \in \mathbb{R}_{++}^n$, their Bregman divergence is the KL-divergence:
$$D_\Phi(x, y) = \textstyle\sum_i (x_i \log(x_i / y_i) - x_i + y_i). \tag{10}$$
In this case, the Bregman projection of $y \in \mathbb{R}_{++}^n$ onto Δ is $\operatorname{proj}_\Delta^\Phi(y) = y / \|y\|_1$ and (8) becomes
$$x^{(k+1)} = x^{(k)} \exp(-\eta_k g^{(k)}) / \|x^{(k)} \exp(-\eta_k g^{(k)})\|_1. \tag{11}$$

Example 2.3 (Negative entropy on Δ)**.** Let $\phi(x) = \sum_i (x_i \log(x_i) - x_i)$ and define $\Phi(x) = \phi(x) + \delta(x|\Delta)$, where $\delta(\cdot|\Delta)$ is the indicator function of Δ, i.e., $\delta(x|\Delta) = 0$ if $x \in \Delta$ and $+\infty$ otherwise. Apparently $\operatorname{dom} \Phi = \Delta$, which has an empty interior. As a result, Φ is not of Legendre type and in fact is not differentiable (see Appendix B). However, the MD algorithm is still well-defined. Specifically, in (7) we interpret $\nabla \Phi(x^{(k)})$ as any subgradient in the subdifferential $\partial \Phi(x^{(k)})$, and find $y^{(k+1)}$ such that there exists some $\nabla \Phi(y^{(k+1)}) \in \partial \Phi(y^{(k+1)})$ to make the equality hold.

Despite $\nabla \Phi(y)$ being multi-valued as a subgradient, the Bregman divergence (5) is still well defined (Corollary B.3). As a result, D_Φ is the same as (10). Using the fact $x, y \in \Delta$, it can be simplified as
$$D_\Phi(x, y) = \textstyle\sum_i x_i \log(x_i / y_i). \tag{12}$$

In addition, we have $\Phi^*(x^*) = \log\left(\sum_i \exp(x_i^*)\right)$ with $\operatorname{dom} \Phi^* = \mathbb{R}^n$ [Rockafellar, 1970, Section 16]. Clearly, Φ^* is a differentiable function throughout $\mathbb{R}^n$ and
$$\nabla \Phi^*(x^*) = \exp(x^*) / \|\exp(x^*)\|_1.$$
In this case, the MD algorithm (8) yields the same update as (11). However, the projection step is no longer needed because the range of $\nabla \Phi^*$ is the interior of Δ and we can simply express MD as
$$x^{(k+1)} = \nabla \Phi^*\big(\nabla \Phi(x^{(k)}) - \eta_k g^{(k)}\big).$$
Remark 2.4. Although Examples 2.2 and 2.3 give the same update (11), there are subtle differences in the theory. In particular, we have $\nabla \Phi^* = (\nabla \Phi)^{-1}$ in Example 2.2 and the equivalence between (8) and 9 is easily established. However, this is not the case for Example 2.3 because Φ is not Legendre and nondifferentiable. Consequently, we can no longer leverage the convex optimization machinery as in the tabular case [e.g., Xiao, 2022, Lan, 2023] for convergence analysis with general function approximation. Instead, we extend the classical MD theory to work with mirror maps whose gradient mapping and conjugate mapping are not inverses of each other; see Appendix B, Lemma B.2.

Algorithm 1 Dual Approximation Policy Optimization (DAPO)

1: **Input:** Initialize policy $\pi^{(0)}$ with parameters $\theta^{(0)}$; mirror map Φ
2: **for** $k = 0, \ldots, K-1$ **do**
3: Find $\widehat{Q}^{(k)}$ that approximates $Q^{(k)}$ (Critic Update)
4: Find $\theta^{(k+1)}$ that (approximately) solves the problem

$$\min_{\theta \in \Theta} \mathbb{E}_{s \sim d_\rho^{(k)}} \Big[D_{\Phi^*}\big(\nabla\Phi(\pi_s^{(k)}) - \eta_k \widehat{Q}_s^{(k)},\, f_s^\theta\big)\Big]$$

5: Assign $\pi_s^{(k+1)} = \mathrm{proj}_{\Delta(\mathcal{A})}^{\Phi}\big(\nabla\Phi^*(f_s^{\theta^{(k+1)}})\big),\ s \in \mathcal{S}$
6: **end for**

3 Policy Optimization with Dual Function Approximation

Recall the setting of MDP in Section 2.1. In the tabular case, the policy mirror descent (PMD) method [Shani et al., 2020a, Lan, 2023, Xiao, 2022] takes the form of (9):

$$\pi_s^{(k+1)} = \operatorname*{arg\,min}_{\pi_s \in \Delta(\mathcal{A})} \Big\{ \eta_k \big\langle \widehat{Q}_s^{(k)}, \pi_s \big\rangle + D_\Phi(\pi_s, \pi_s^{(k)}) \Big\}, \quad s \in \mathcal{S}, \tag{13}$$

where $\widehat{Q}^{(k)}$ is some approximation of $Q^{(k)}$. We note that $Q^{(k)}$ is not the gradient of the value function V_ρ^π at $\pi^{(k)}$, which is given in (4); rather, it is a preconditioned gradient [Kakade, 2001].

When the size of the state-action space becomes large (possibly infinite), we have to resort to function approximation. Specifically, let π^θ be a differentiable mapping from the set of parameters $\Theta \subset \mathbb{R}^n$ to the set of stochastic policies. The parameter update step corresponding to (13) becomes

$$\theta^{(k+1)} = \operatorname*{arg\,min}_{\theta \in \Theta} \mathbb{E}_{s \sim d_\rho^{(k)}} \Big[\mathbb{E}_{a \sim \pi_s^\theta} \big[\widehat{Q}_{s,a}^{(k)}\big] + D_\Phi(\pi_s^\theta, \pi_s^{(k)}) \Big], \tag{14}$$

where $\pi^{(k)}$ means $\pi^{\theta^{(k)}}$, and $d_\rho^{(k)}$ and $\widehat{Q}_{s,a}^{(k)}$ are simple notations for $d_\rho^{\pi^{(k)}}$ and $\widehat{Q}_{s,a}^{\pi^{(k)}}$ respectively. This approach is adopted by, e.g., Tomar et al. [2020] and Vaswani et al. [2021]. However, the optimization problem is no longer convex in θ, and its convergence analysis becomes more challenging.

Alfano et al. [2023] introduced Approximate Mirror Policy Optimization (AMPO), a framework that incorporates general parametrization into PMD with convergence guarantees. A key instrument they introduced is the *Bregman projected policy* class. The idea is to use a parametrized function $f^\theta : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ to approximate the dual update in (7), which in the context of PMD is $\nabla\Phi(\pi^{(k)}) - \eta_k \widehat{Q}^{(k)}$. Then follow the second step in MD to define the policy class

$$\big\{\pi^\theta : \pi_s^\theta = \mathrm{proj}_{\Delta(\mathcal{A})}^{\Phi}\big(\nabla\Phi^*(f_s^\theta)\big),\ s \in \mathcal{S}\big\}, \quad \theta \in \Theta.$$

For example, using the negative-entropy (Example 2.2 or 2.3), it leads to the softmax policy class:

$$\pi_{s,a}^\theta = \exp(f_{s,a}^\theta)/\|\exp(f_s^\theta)\|_1, \quad (s,a) \in \mathcal{S} \times \mathcal{A}. \tag{15}$$

While such policy classes are widely used in both theory and practice, recognizing them as the composition of a Bregman projection, a conjugate mirror map and a generic function approximation f^θ (such as neural networks) allows more structured and sharper convergence analysis.

Facilitated with the Bregman projected policy class, extending PMD with function approximation rests upon how we approximate $\nabla\Phi(\pi^{(k)}) - \eta_k \widehat{Q}^{(k)}$ (*existing in the dual space*) using f^θ. AMPO [Alfano et al., 2023] proposes to minimize the expected L_2-distance between them, i.e.,

$$\min_{\theta} \mathbb{E}_{s \sim d_\rho^{(k)}} \Big[\big\| f_s^\theta - \big(\nabla\Phi(\pi_s^{(k)}) - \eta_k \widehat{Q}_s^{(k)}\big) \big\|_2^2 \Big]. \tag{16}$$

On the other hand, Lan [2022] tries to minimize the expected (in state distribution) L_∞-norm of the difference between f_s^θ and $\nabla\Phi(\pi_s^{(k)}) - \eta \widehat{Q}_s^{(k)}$.

In contrast, we propose to use the corresponding *dual Bregman divergence* D_{Φ^*} to measure their similarity in the dual space. In particular, Our method finds $\theta^{(k+1)}$ by (approximately) solving

$$\min_{\theta \in \Theta} \mathbb{E}_{s \sim d_\rho^{(k)}} \Big[D_{\Phi^*}\big(\nabla\Phi(\pi_s^{(k)}) - \eta_k \widehat{Q}_s^{(k)},\, f_s^\theta\big)\Big], \tag{17}$$

where $d_\rho^{(k)}$ can be replaced with other distributions to accommodate the scenario of off-policy training. Here, we can see that the similarity between the two dual vectors f_s^θ and $\nabla\Phi(\pi_s^{(k)}) - \eta\widehat{Q}_s^{(k)}$ are measured by the Bregman divergence of Φ^*, which naturally lives in the dual space. Together with the Bregman divergence of Φ used in policy projection, they form a complete duality framework. A complete description of our method is given as Algorithm 1, and we call it Dual Approximation Policy Optimization (DAPO).

3.1 Instantiations of DAPO

We give three instantiations of DAPO using the three mirror maps given in Examples 2.1-2.3. In deriving these instantiations as well as implementing the algorithms, instead of directly using the dual Bregman divergence D_{Φ^*}, it is often more convenient to use the following identity:

$$D_{\Phi^*}\big(\nabla\Phi(\pi_s^{(k)}) - \eta_k\widehat{Q}_s^{(k)},\, f_s^\theta\big) = D_\Phi\big(\nabla\Phi^*(f_s^\theta),\, \nabla\Phi^*(\nabla\Phi(\pi_s^{(k)}) - \eta_k\widehat{Q}_s^{(k)})\big). \tag{18}$$

See Corollary B.4 for a proof. This identity will also facilitate our convergence analysis later.

DAPO-L_2. With Φ being the squared L_2-norm mirror map described in Example 2.1, the approximation problem in (17) (same as line 4 in Algorithm 1) becomes

$$\min_\theta\ \mathbb{E}_{s\sim d_\rho^{(k)}}\left[\big\|f_s^\theta - \pi_s^{(k)} + \eta_k\widehat{Q}_s^{(k)}\big\|_2^2\right], \tag{19}$$

and Line 5 of Algorithm 1 is the Euclidean projection

$$\pi_s^{(k+1)} = \operatorname*{arg\,min}_{\pi\in\Delta(\mathcal{A})}\big\|\pi - f_s^{(k+1)}\big\|_2^2.$$

Here we have used the simpler notation $f_s^{(k+1)}$ for $f_s^{\theta^{(k+1)}}$.

DAPO-KL*. With Φ being the negative entropy defined on $\mathbb{R}_+^{|\mathcal{A}|}$ (see Example 2.2), we have

$$\nabla\Phi^*(f_s^{(k+1)}) = \exp(f_{s,a}^\theta),$$
$$\nabla\Phi^*\big(\nabla\Phi(\pi^{(k)}) - \eta_k\widehat{Q}^{(k)}\big) = \pi_s^{(k)}\exp\big(-\eta_k\widehat{Q}_s^{(k)}\big).$$

Using the identity (18), we can write the loss in the approximation problem (17) as

$$\mathbb{E}_{s\sim d_\rho^{(k)}}\left[D_{\mathrm{KL}}\left(\exp(f_s^\theta)\big\|\pi_s^{(k)}\exp\big(-\eta_k\widehat{Q}_s^{(k)}\big)\right)\right], \tag{20}$$

where D_{KL} is given by (10). Policy projection as in (15) is necessary to obtain $\pi^{(k+1)}$ because $\nabla\Phi^*(f_s^{(k+1)})$ is not in the simplex in general. DAPO-KL* has disadvantages in both theory and practice compared with its close variant DAPO-KL, which we will explain next.

DAPO-KL. With Φ being the negative entropy restricted on $\Delta(\mathcal{A})$ (see Example 2.3), the range of $\nabla\Phi^*$ is $\Delta(\mathcal{A})$, thus the projection step (Line 5 of Algorithm 1) becomes redundant. Here, we have

$$\pi_s^{(k+1)} = \nabla\Phi^*(f_s^{(k+1)}) = \exp(f_{s,a}^\theta)/\|\exp(f_s^\theta)\|_1,$$
$$\nabla\Phi^*\big(\nabla\Phi(\pi^{(k)}) - \eta_k\widehat{Q}^{(k)}\big) = \pi_s^{(k)}\exp\big(-\eta_k\widehat{Q}_s^{(k)}\big)/Z_s^{(k)},$$

where $Z_s^{(k)} = \big\|\pi_s^{(k)}\exp\big(-\eta_k\widehat{Q}_s^{(k)}\big)\big\|_1$. Again using (18), the loss in (17) becomes

$$\mathbb{E}_{s\sim d_\rho^{(k)}}\left[D_{\mathrm{KL}}\left(\pi_s^\theta\ \Big\|\ \pi_s^{(k)}\exp\big(-\eta_k\widehat{Q}_s^{(k)}\big)/Z_s^{(k)}\right)\right], \tag{21}$$

where D_{KL} is given by (12). There are several distinctions between DAPO-KL and DAPO-KL*.

- The approximation loss in (21) is in terms of the full policy parametrization π^θ (normalized over the simplex), matching the implementation of several popular algorithms [Tomar et al., 2020, Vaswani et al., 2021]. In contrast, the loss in (20) is in terms of the unnormalized entity $\exp(f^\theta)$, which will suffer additional loss after policy projection.
- In theory,we are able to provide a competitive convergence analysis of DAPO-KL (see Section 4.1) thanks to the fact that the two arguments in D_{KL} in (21) are both on the simplex, which is not the case in (20).

For these reasons, we will only consider DAPO-KL from now on. However, we think it is necessary to expose the subtleties between the two variants, because many works on policy mirror descent methods [e.g. Alfano et al., 2023], assumes $\nabla\Phi^* = (\nabla\Phi)^{-1}$. We demonstrate that the more nuanced extension of the MD theory (Lemma B.2) is crucial for developing and analyzing practical algorithms.

3.2 Comparison with AMPO, MDPO and FMA-PG

AMPO [Alfano et al., 2023] replaces the minimization problem in Line 4 of Algorithm 1 by (16) regardless of the mirror map used in policy projection. More concretely, let Φ_1 be the negative entropy on $\mathbb{R}_+^n$ and Φ_2 be the squared L_2 norm. Then AMPO's approximation loss can be written as

$$\mathbb{E}_{s\sim d_\rho^{(k)}}\left[D_{\Phi_2^*}\left(\nabla\Phi_1(\pi_s^{(k)}) - \eta_k\widehat{Q}_s^{(k)}, f_s^\theta\right)\right], \tag{22}$$

and Φ_1 is again used in the policy projection step. In theory, as long as the approximation error is small, it is possible to establish convergence of the method [Alfano et al., 2023]. However, such a mismatch, or inconsistency, between approximations in primal and dual spaces may cause problems when the approximation error cannot be made sufficiently small. This is precisely the case in practice, where we can only afford to run at most a few steps of the stochastic gradient method to reduce the approximation error. The importance of the consistency between the two mirror maps has also been pointed out by Tomar et al. [2020].

In Section 5, we demonstrate that on standard benchmarks DAPO-KL obtains state-of-the-art performance with only one step of stochastic gradient method in reducing the approximation loss (17), comparable to SAC [Haarnoja et al., 2018b]. On the other hand, we could not get AMPO competitive with many numbers of stochastic gradient steps.

The Mirror Descent Policy Optimization (MDPO) method of Tomar et al. [2020] is based on minimizing over θ directly in the formulation (14). If π^θ belongs to the softmax class of (15) and D_Φ is the KL-divergence, then it is equivalent to DAPO-KL. Therefore, our convergence analysis in Section 4.1 directly applies to MDPO, which is not provided by Tomar et al. [2020].

The Functional Mirror Ascent (FMA-PG) framework of Vaswani et al. [2021] also takes the form (14). However, similar to MDPO, Vaswani et al. [2021] did not exploit any composition structure of the parametrization π^θ or the MDP structure. Rather, they conducted convergence analysis based on the general theory for smooth, non-convex optimization, which leads to considerably weaker results.

3.3 SAC as a special case of DAPO-KL

Soft Actor-Critic (SAC) [Haarnoja et al., 2018a] is a very popular reinforcement learning algorithm, which was developed under the framework of entropy-regularized reinforcement learning. Tomar et al. [2020] compared SAC's actor update loss function with (21) and pointed out that SAC is similar to MDPO (same as DAPO-KL, as we discussed above) by replacing the previous iterate $\pi^{(k)}$ with the uniform distribution. Here, we will prove a much stronger result, showing that by choosing the learning rate η_k appropriately, Equation (21) become exactly SAC's policy update rule.

To prove this, we will first briefly introduce the framework of entropy-regularized reinforcement learning and then derive the corresponding DAPO-KL algorithm under this framework. [1] With a regularization parameter $\tau > 0$, the regularized value function in this framework is

$$V_{\tau,\rho}^\pi = \mathbb{E}_{a_t\sim\pi_{s_t}}\left[\sum_{t=0}^\infty \gamma^t\left(c(s_t,a_t)+\tau\log\pi(a_t\mid s_t)\right)\mid s_0\sim\rho\right].$$

Then, we can similarly define the Q-value function as

$$Q_\tau^\pi(s,a) = \mathbb{E}_{a_t\sim\pi_{s_t}}\left[\sum_{t=0}^\infty \gamma^t\left(c(s_t,a_t)+\tau\log\pi(a_t\mid s_t)\right)\mid s_0=s, a_0=a\right]. \tag{23}$$

As shown in Cayci et al. [2021], the policy gradient for this entropy-regularized value function is

$$\nabla_s V_{\tau,\rho}^\pi = \tfrac{1}{1-\gamma} d_{\rho,s}^\pi Q_{\tau,s}^\pi \in \mathbb{R}^{|\mathcal{A}|}.$$

Therefore, we can obtain the corresponding DAPO algorithm by using this policy gradient. Setting Φ as the negative entropy, we obtain the corresponding DAPO-KL update rule as

$$\theta^{(k+1)} \in \arg\min_\theta \mathbb{E}_{s\sim d_\rho^{(k)}}\left[D_{\mathrm{KL}}\left(\pi_s^\theta \,\middle\|\, \pi_s^{(k)}\exp\left(-\eta_k Q_{\tau,s}^{(k)}\right)/Z_s^{(k)}\right)\right], \tag{24}$$

where $Z_s^{(k)}$ is the normalization factor and we take the exact Q-value function for simplicity.

[1] See Cen et al. [2022] and Cayci et al. [2021] for backgrounds on entropy-regularized reinforcement learning.

Now, we switch our attention to the SAC algorithm in Haarnoja et al. [2018a]. The subtlety is that the soft Q-value function used in Haarnoja et al. [2018a] is defined differently from the one in equation (23). To be more clear, we denote it as q_τ^π, which for any $(s,a) \in \mathcal{S} \times \mathcal{A}$ is defined as

$$\begin{cases} q_\tau^\pi(s,a) = c(s,a) + \gamma \mathbb{E}_{s' \sim \mathcal{P}(s,a)}\left[V_\tau^\pi(s')\right], \\ V_\tau^\pi(s) = \mathbb{E}_{a' \sim \pi_s}\left[\tau \log \pi(a' \mid s) + q_\tau^\pi(s,a')\right]. \end{cases} \tag{25}$$

Note that the definition of V_τ^π remains unaffected. Then, we can immediately obtain the relation $q_\tau^\pi(s,a) = Q_\tau^\pi(s,a) - \tau \log \pi(a \mid s)$. As a result, the policy update rule in SAC is[2]

$$\begin{aligned} \theta^{(k+1)} &\in \arg\min_\theta \mathbb{E}_{s \sim d_\rho^{(k)}}\left[D_{\mathrm{KL}}\left(\pi_s^\theta \,\middle\|\, \exp\left(-q_{\tau,s}^{(k)}/\tau\right)/Z_s^{(k)}\right)\right] \\ &= \arg\min_\theta \mathbb{E}_{s \sim d_\rho^{(k)}}\left[D_{\mathrm{KL}}\left(\pi_s^\theta \,\middle\|\, \exp\left(-Q_{\tau,s}^{(k)}/\tau + \log \pi_s^{(k)}\right)/Z_s^{(k)}\right)\right] \\ &= \arg\min_\theta \mathbb{E}_{s \sim d_\rho^{(k)}}\left[D_{\mathrm{KL}}\left(\pi_s^\theta \,\middle\|\, \pi_s^{(k)} \exp\left(-Q_{\tau,s}^{(k)}/\tau\right)/Z_s^{(k)}\right)\right], \end{aligned}$$

which is exactly the same as the update rule in Eq. (24) if we take $\eta_k = \frac{1}{\tau}$ for any k. Therefore, we conclude that *SAC's update rule can be obtained by taking $\eta_k = \frac{1}{\tau}$ for any k in DAPO-KL for an entropy-regularized MDP.* As an immediate consequence, we can have a tight convergence rate analysis for SAC, as given in Section 4.2.

4 Convergence Analysis

In this section, we present the convergence analysis of DAPO-KL and SAC. These results are nontrivial extensions of similar results for PMD method in the tabular case [Xiao, 2022] and with the log-linear policy class [Yuan et al., 2022]. The analysis of DAPO-L_2 is deferred to Appendix C.

4.1 Analysis of DAPO-KL

We make following three assumptions about running Algorithm 1, with initial distribution $\rho \in \Delta(\mathcal{S})$.

(A1) There exist constants $\epsilon_{\mathrm{critic}}, \epsilon_{\mathrm{actor}} > 0$ such that for every iteration k, it holds that

$$\begin{aligned} &\mathbb{E}_{s \sim d_\rho^{(k)}}\left[\left\|\widehat{Q}_s^{(k)} - Q_s^{(k)}\right\|_\infty\right] \le \epsilon_{\mathrm{critic}}, \\ &\mathbb{E}_{s \sim d_\rho^{(k)}}\left[D_{\mathrm{KL}}\left(\pi_s^{(k+1)} \,\middle\|\, \pi_s^{(k)} \exp\left(-\eta_k \widehat{Q}_s^{(k)}\right)/Z_s^{(k)}\right)\right] \le \eta_k \epsilon_{\mathrm{actor}} \end{aligned} \tag{26}$$

(A2) There exists a constant $\vartheta_\rho \ge 1$ such that for any k,

$$\max\left\{\left\|\frac{d_\rho^\star}{d_\rho^{(k+1)}}\right\|_\infty, \left\|\frac{d_\rho^{(k+1)}}{d_\rho^{(k)}}\right\|_\infty, \left\|\frac{d_{d_\rho^\star}^{(k+1)}}{d_\rho^{(k)}}\right\|_\infty, \left\|\frac{d_{d_\rho^\star}^{(k+1)}}{d_\rho^\star}\right\|_\infty\right\} \le \vartheta_\rho.$$

(A3) There exists a constant $C_\rho > 0$ such that for any k,

$$\max_{s \in \mathrm{supp}\left(d_\rho^{(k)}\right)}\left\{\left\|\frac{\pi_s^\star}{\pi_s^{(k+1)}}\right\|_\infty, \left\|\frac{\pi_s^{(k)}}{\pi_s^{(k+1)}}\right\|_\infty\right\} \le C_\rho.$$

Here, in (A1), we assume that $\widehat{Q}^{(k)}$ is a good enough approximation of $Q^{(k)}$, which is a problem that has been extensively studied both theoretically and empirically [Li and Lan, 2023, Chen et al., 2022a, Fujimoto et al., 2018]. We also assume that the parameterized function f^θ is powerful enough to approximate the dual vector $\nabla\Phi(\pi^{(k)}) - \eta_k \widehat{Q}^{(k)}$, which is a common assumption for studying function approximation [Alfano et al., 2023, Lan, 2022, Agarwal et al., 2021].

Then, (A2) assumes that the distribution mismatch coefficient is bounded, which is often needed for analyzing policy gradient methods [Xiao, 2022, Yuan et al., 2022] and can be satisfied if we take

[2] The original proposition of SAC in Haarnoja et al. [2018a] uses $\exp(q_{\tau,s}^{(k)}/\tau)$ instead of $\exp(-q_{\tau,s}^{(k)}/\tau)$ because it considers reward maximization instead of cost minimization.

$\rho = \mathsf{Unif}(\mathcal{S})$ (see Lemma D.7). Meanwhile, (A3) is an assumption on policy evolution. It holds, for example, when we apply DAPO-KL with entropy regularization [Cayci et al., 2021, Cen et al., 2022] and it covers the case of SAC (see discussion in Section 3.2).

Under these assumptions, we have the following theorem and the proof is given in Appendix D.

Theorem 4.1 (Linear Convergence of DAPO-KL)**.** *Consider Algorithm 1 with initial policy $\pi^{(0)}$, initial distribution $\rho \in \Delta(\mathcal{S})$ and Φ being the negative entropy restricted on $\Delta(\mathcal{A})$. Suppose Assumptions (A1), (A2) and (A3) hold and the step sizes satisfy $\eta_0 > 1$ and $\eta_{k+1} \geq (\vartheta_\rho/(\vartheta_\rho - 1))\,\eta_k$ for all $k \geq 0$. Then, for any comparator policy $\pi^\star$, we have*

$$V_\rho^{(K)} - V_\rho^\star \leq \left(1 - \frac{1}{\vartheta_\rho}\right)^K \left(V_\rho^{(0)} - V_\rho^\star + \frac{D_0^\star/(\vartheta_\rho - 1)}{(1-\gamma)\eta_0}\right) + \frac{\vartheta_\rho^2 \psi(\epsilon_{\text{actor}}) + 2\vartheta_\rho \epsilon_{\text{critic}}}{1-\gamma},$$

where $\psi(x) = (1 + C_\rho)\left(x + \sqrt{2x}\right)$ for $x \geq 0$.

Remark 4.2. Theorem 4.1 obtains linear convergence (up to an error floor dictated by ϵ_{actor} and ϵ_{critic}) by employing a geometrically growing learning rate. We can also obtain $O(1/K)$ sublinear rates with a constant step size following similar proof techniques. We omit the details as such results can be found in, for example, Xiao [2022], Yuan et al. [2022], Alfano et al. [2023].

4.2 Analysis of SAC

Although we have shown that SAC is a special case of DAPO-KL in Section 3.2, its convergence analysis is somewhat different from the above, as it is formulated under the framework of entropy-regularized reinforcement learning. Specifically, the key difference in analysis lies in the following modified performance difference lemma.

Lemma 4.3 (Modified Performance Difference Lemma)**.** *For any two policies $\pi, \tilde{\pi} : \mathcal{S} \mapsto \Delta(\mathcal{A})$, initial distribution $\rho \in \Delta(\mathcal{S})$ and regularization strength $\tau > 0$, it holds that*

$$\begin{aligned} V_{\tau,\rho}^{\pi} - V_{\tau,\rho}^{\tilde{\pi}} &= \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_\rho^\pi} \left[\left\langle Q_{\tau,s}^{\tilde{\pi}}, \pi_s - \tilde{\pi}_s \right\rangle + \tau D_{\text{KL}}(\pi_s \| \tilde{\pi}_s) \right] \\ &= \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_\rho^{\tilde{\pi}}} \left[\left\langle Q_{\tau,s}^{\pi}, \pi_s - \tilde{\pi}_s \right\rangle - \tau D_{\text{KL}}(\tilde{\pi}_s \| \pi_s) \right]. \end{aligned}$$

The proof is given in Appendix E. Using Lemma 4.3, the convergence guarantee of DAPO-KL with entropy regularization (and thus SAC) is summarized in the following theorem.

Theorem 4.4 (Sublinear Convergence of SAC)**.** *Consider running Algorithm 1 for entropy-regularized reinforcement learning with initial policy $\pi^{(0)}$, regularization strength τ, initial distribution $\rho \in \Delta(\mathcal{S})$ and Φ being the negative entropy restricted on $\Delta(\mathcal{A})$. Suppose Assumptions (A1), (A2) and (A3) hold and the step sizes satisfy $\eta_k = \eta \leq \frac{1}{\tau \vartheta_\rho}$ for any k. Then, for any comparator policy $\pi^\star$, we have*

$$\frac{1}{K} \sum_{k=0}^{K-1} \left(V_{\tau,\rho}^{(k)} - V_{\tau,\rho}^\star\right) \;\leq\; \frac{1}{K}\left(\frac{D_0^\star}{(1-\gamma)\eta} + \frac{V_{\tau,d_\rho^\star}^{(0)}}{1-\gamma}\right) + \frac{\vartheta_\rho \psi(\epsilon_{\text{actor}}) + (2-\gamma)\vartheta_\rho \epsilon_{\text{critic}}}{(1-\gamma)^2},$$

where $D_0^\star = \mathbb{E}_{s \sim d_\rho^\star}\left[D_{\text{KL}}\left(\pi_s^\star \parallel \pi_s^{(0)}\right)\right]$ and $\psi(x) = (1 + C_\rho)\left(x + \sqrt{2x}\right)$ for $x \geq 0$.

The full proof is given in Appendix E. To the best of our knowledge, this is the first convergence rate analysis of the SAC algorithm under general function approximation.

5 Experiments

In this section, we present our experiment results on several standard MuJoCo benchmark tasks [Todorov et al., 2012]. We compare the performance of DAPO-KL, SAC [Haarnoja et al., 2018b], and AMPO [Alfano et al., 2023].

In order to demonstrate the importance of having primal-dual consistency, we modify AMPO to enforce it albeit in a naive way. Specifically, as discussed in Section 3.2, AMPO's approximation loss

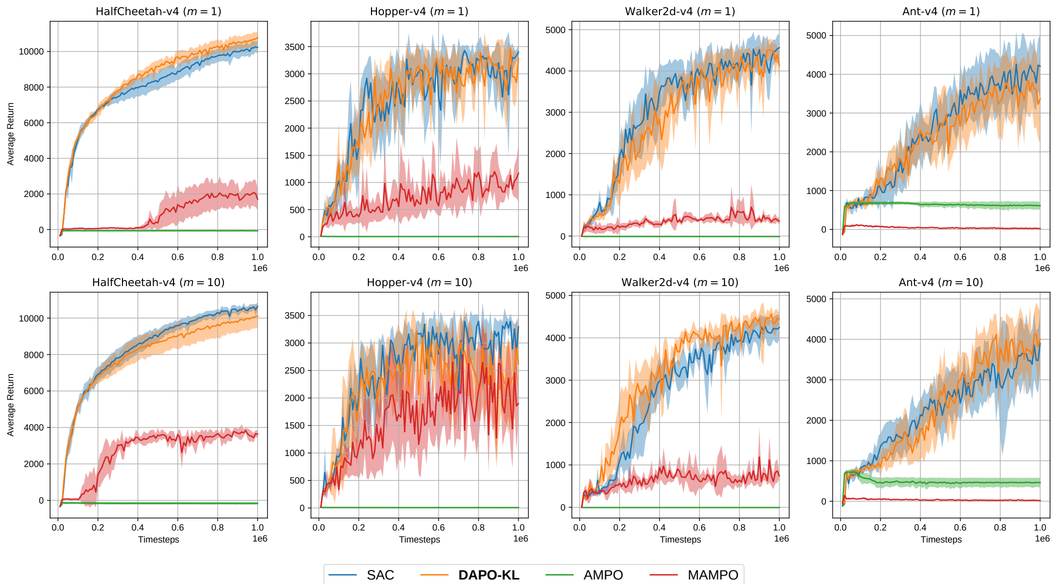

Figure 1: Average return curves on MuJoCo benchmarks. Each curve is averaged over 5 random seeds and the shaded area represents the 95% confidence interval. Here m represents the number of stochastic gradient steps in each policy update iteration.

can be expressed as (22) with Φ_1 being negative entropy and Φ_2 being squared L_2-norm. Therefore, a naive way to enforce primal-dual consistency is to replace Φ_1 by Φ_2 in (22), which gives

$$\mathbb{E}_{s\sim d_\rho^{(k)}}\left[\left\| f_s^\theta - \left(\pi_s^{(k)} - \eta_k \widehat{Q}_s^{(k)}\right)\right\|_2^2\right]. \tag{27}$$

We call this algorithm Modified AMPO (MAMPO). Note that in MAMPO, Φ_1 (negative entropy) is still used in the policy projection step, which is different from DAPO-L_2.

Although in theory we assume that the policy optimization loss is approximately minimized in each iteration, in practice, it may be only feasible to run a few steps of the stochastic gradient method to reduce the loss. Therefore, the number of stochastic gradient steps per iteration can be an important hyper-parameter for the algorithm. In experiments, all algorithms are evaluated under both $m = 1$ and $m = 10$ stochastic gradient step per iteration. Implementation details are given in Appendix F.

The results are summarized in Fig. 1. From the plots, we can see that DAPO-KL performs about the same as SAC on all tasks, which is expected as we have shown that SAC is a special case of DAPO-KL. Meanwhile, they are not sensitive to number of stochastic gradient steps per iteration.

On the other hand, AMPO fails to learn anything non-trivial on all tasks no matter it uses $m = 1$ or $m = 10$ stochastic gradient steps. Nevertheless, we retain the possibility that our implementation of AMPO may not be the optimal and provide more details of its hyperparameter tuning in Appendix G. In contrast, MAMPO is able to complete non-trivial learning among three tasks and gets better with more gradient steps, indicating the benefit of the primal-dual consistency in (27). However, it is still far inferior to DAPO-KL and SAC.

6 Conclusions

DAPO is a novel duality framework for incorporating general function approximation into policy mirror descent methods. Besides the mirror map in policy projection, it uses the dual mirror map for measuring the function approximation error. We establish linear and sublinear convergence rates of DAPO under different step size rules and show that it incorporates state-of-the-art algorithms like SAC as a special case, immediately providing them with strong convergence guarantees.

For future directions, DAPO paves the way for exploring new variants of PMD methods based on different mirror maps, e.g., with the negative Tsallis entropy. Another interesting question to investigate is how to characterize the effects of using inconsistent mirror maps in AMPO.

References

Alekh Agarwal, Mikael Henaff, Sham Kakade, and Wen Sun. Pc-pg: Policy cover directed exploration for provable policy gradient learning. *Advances in neural information processing systems*, 33: 13399–13412, 2020.

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

Carlo Alfano, Rui Yuan, and Patrick Rebeschini. A novel framework for policy mirror descent with general parametrization and linear convergence. *arXiv preprint arXiv:2301.13139*, 2023.

Pierre-Cyril Aubin-Frankowski, Anna Korba, and Flavien Léger. Mirror descent with relative smoothness in measure spaces, with application to sinkhorn and em. *arXiv preprint arXiv:2206.08873*, 2022.

Heinz G. Bauschke and Jonathan M. Borwein. Legendre functions and the method of random bregman projections. *Journal of Convex Analysis*, 4(1):27–67, 1997.

Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

Dimitri Bertsekas. *Convex optimization theory*, volume 1. Athena Scientific, 2009.

Dimitri P Bertsekas. Dynamic programming and optimal control 4th edition, volume ii. *Athena Scientific*, 2015.

Jalaj Bhandari and Daniel Russo. On the linear convergence of policy gradient methods for finite mdps. In *International Conference on Artificial Intelligence and Statistics*, pages 2386–2394. PMLR, 2021.

Sébastien Bubeck. *Convex Optimization: Algorithms and Complexity*. Number 8:3-4 in Foundations and Trends in Machine Learning. now Publishers Inc., 2015.

Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

Qi Cai, Zhuoran Yang, Chi Jin, and Zhaoran Wang. Provably efficient exploration in policy optimization. In *International Conference on Machine Learning*, pages 1283–1294. PMLR, 2020.

Semih Cayci, Niao He, and Rayadurgam Srikant. Linear convergence of entropy-regularized natural policy gradient with linear function approximation. *arXiv preprint arXiv:2106.04096*, 2021.

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.

Zaiwei Chen and Siva Theja Maguluri. Sample complexity of policy-based methods under off-policy sampling and linear function approximation. In *International Conference on Artificial Intelligence and Statistics*, pages 11195–11214. PMLR, 2022.

Zaiwei Chen, John Paul Clarke, and Siva Theja Maguluri. Target network and truncation overcome the deadly triad in q-learning. *arXiv preprint arXiv:2203.02628*, 2022a.

Zaiwei Chen, Sajad Khodadadian, and Siva Theja Maguluri. Finite-sample analysis of off-policy natural actor–critic with linear function approximation. *IEEE Control Systems Letters*, 6:2611–2616, 2022b.

Casey Chu, Jose Blanchet, and Peter Glynn. Probability functional descent: A unifying perspective on gans, variational inference, and reinforcement learning. In *International Conference on Machine Learning*, pages 1213–1222. PMLR, 2019.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018.

Jakub Grudzien, Christian A Schroeder De Witt, and Jakob Foerster. Mirror learning: A unifying framework of policy optimisation. In *International Conference on Machine Learning*, pages 7825–7844. PMLR, 2022.

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, 2018a.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018b.

Emmeran Johnson, Ciara Pike-Burke, and Patrick Rebeschini. Optimal convergence rate for exact policy mirror descent in discounted markov decision processes. *arXiv preprint arXiv:2302.11381*, 2023.

Sham M Kakade. A natural policy gradient. *Advances in neural information processing systems*, 14, 2001.

Sajad Khodadadian, Prakirt Raj Jhunjhunwala, Sushil Mahavir Varma, and Siva Theja Maguluri. On the linear convergence of natural policy gradient algorithm. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 3794–3799. IEEE, 2021.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Vijay Konda and John Tsitsiklis. Actor-critic algorithms. *Advances in neural information processing systems*, 12, 1999.

Guanghui Lan. Policy optimization over general state and action spaces. *arXiv preprint arXiv:2211.16715*, 2022.

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

Guanghui Lan, Yan Li, and Tuo Zhao. Block policy mirror descent. *SIAM Journal on Optimization*, 33(3):2341–2378, 2023.

Yan Li and Guanghui Lan. Policy mirror descent inherently explores action space. *arXiv preprint arXiv:2303.04386*, 2023.

Yan Li, Guanghui Lan, and Tuo Zhao. Homotopic policy mirror descent: Policy convergence, implicit regularization, and improved sample complexity. *arXiv preprint arXiv:2201.09457*, 2022.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Boyi Liu, Qi Cai, Zhuoran Yang, and Zhaoran Wang. Neural proximal/trust region policy optimization attains globally optimal policy. *arXiv preprint arXiv:1906.10306*, 2019.

Jincheng Mei, Bo Dai, Alekh Agarwal, Mohammad Ghavamzadeh, Csaba Szepesvári, and Dale Schuurmans. Ordering-based conditions for global convergence of policy gradient methods. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

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.

Arkadi Semenovič Nemirovski and David Borisovich Yudin. Problem complexity and method efficiency in optimization. 1983.

Gergely Neu, Anders Jonsson, and Vicenç Gómez. A unified view of entropy-regularized markov decision processes. *arXiv preprint arXiv:1705.07798*, 2017.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 1994.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *The Journal of Machine Learning Research*, 22(1):12348–12355, 2021.

R. Tyrrell Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

Ralph Tyrrell Rockafellar. Conjugates and legendre transforms of convex functions. *Canadian Journal of Mathematics*, 19:200–205, 1967.

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.

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

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, pages 5668–5675, 2020a.

Lior Shani, Yonathan Efroni, Aviv Rosenberg, and Shie Mannor. Optimistic policy optimization with bandit feedback. In *International Conference on Machine Learning*, pages 8604–8613. PMLR, 2020b.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

Manan Tomar, Lior Shani, Yonathan Efroni, and Mohammad Ghavamzadeh. Mirror descent policy optimization. *arXiv preprint arXiv:2005.09814*, 2020.

Sharan Vaswani, Olivier Bachem, Simone Totaro, Robert Müller, Shivam Garg, Matthieu Geist, Marlos C Machado, Pablo Samuel Castro, and Nicolas Le Roux. A general class of surrogate functions for stable and efficient reinforcement learning. *arXiv preprint arXiv:2108.05828*, 2021.

Nino Vieillard, Tadashi Kozuno, Bruno Scherrer, Olivier Pietquin, Rémi Munos, and Matthieu Geist. Leverage the average: an analysis of kl regularization in rl. *arXiv preprint arXiv:2003.14089*, 2020.

Lingxiao Wang, Qi Cai, Zhuoran Yang, and Zhaoran Wang. Neural policy gradient methods: Global optimality and rates of convergence. *arXiv preprint arXiv:1909.01150*, 2019.

Christopher John Cornish Hellaby Watkins. Learning from delayed rewards. 1989.

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

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

Tengyu Xu, Zhe Wang, and Yingbin Liang. Improving sample complexity bounds for (natural) actor-critic algorithms. *Advances in Neural Information Processing Systems*, 33:4358–4369, 2020.

Rui Yuan, Simon S Du, Robert M Gower, Alessandro Lazaric, and Lin Xiao. Linear convergence of natural policy gradient methods with log-linear policies. *arXiv preprint arXiv:2210.01400*, 2022.

Andrea Zanette, Ching-An Cheng, and Alekh Agarwal. Cautiously optimistic policy optimization and exploration with linear function approximation. In *Conference on Learning Theory*, pages 4473–4525. PMLR, 2021.

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.

A Related Work

PG and PMD in tabular MDPs. Although the proposal of policy gradient theorem and natural policy gradient (NPG) can be traced back to around 2000s or even before [Williams, 1992, Konda and Tsitsiklis, 1999, Sutton et al., 1999, Kakade, 2001], the study of its convergence to the global optimum only started in recent years. On the other hand, mirror descent algorithm [Nemirovski and Yudin, 1983] has been extensively studied for a long time as an online learning algorithm Bubeck et al. [2012]. To connect these two, Neu et al. [2017] first shows that NPG can be viewed as a special case of policy mirror descent (PMD) and most of the following convergence analyses are based on this viewpoint. For tabular MDPs, Shani et al. [2020a] shows that unregularized NPG with a softmax policy has a $O(1/\sqrt{K})$ convergence rate. Agarwal et al. [2021], Vieillard et al. [2020], Xu et al. [2020] then improve it to the $O(1/K)$ convergence rate under different settings. After that, Khodadadian et al. [2021], Bhandari and Russo [2021], Xiao [2022] prove the linear convergence rate for the NPG method. Very recently, Johnson et al. [2023] shows that a linear convergence rate is optimal for NPG in tabular MDPs and Mei et al. [2023] provides a new perspective by proving a necessary and sufficient ordering-based condition for NPG convergence in bandit setting.

PG and PMD in regularized MDPs. Another parallel line of work analyzes applying NPG method to maximum entropy reinforcement learning. Cayci et al. [2021], Cen et al. [2022] show that NPG with softmax policies can converge linearly in entropy-regularized MDPs while Lan [2023] also shows general PMD method converges linearly. Then, the linear convergence of PMD is extended to MDPs with general convex regularizers by Zhan et al. [2023]. Meanwhile, Li et al. [2022] and Lan et al. [2023] also propose other variants of PMD methods that converge linearly in entropy-regularized MDPs.

PG and PMD with function approximation. Agarwal et al. [2021] shows Q-NPG with log-linear policies achieves $O(1/\sqrt{K})$ convergence rate while Cayci et al. [2021] and Yuan et al. [2022] show that NPG with log-linear policies can converge linearly in entropy-regularized MDPs and unregularized MDPs. Meanwhile, Chen et al. [2022b] and Chen and Maguluri [2022] show similar $O(1/K)$ and linear convergence result under different assumptions, respectively. For more general function approximation setting, Wang et al. [2019] shows that NPG with two-layer neural network has $O(1/\sqrt{K})$ convergence rate and Liu et al. [2019] shows that NPG with multi-layer neural network achieves $O(1/\sqrt{K})$ convergence rate. Recently, Alfano et al. [2023] shows PMD method with general function approximation can converge linearly. The main difference between Alfano et al. [2023] and our work lies on how we define approximation, as discussed in Section 3.2.

Applications of PG. Together with the rise of deep Q-learning [Mnih et al., 2013], PG methods have also inspired many successful practical algorithms for real-world control task, including DDPG in Lillicrap et al. [2015], TRPO in Schulman et al. [2015], PPO in Schulman et al. [2017] and SAC in Haarnoja et al. [2018b,a]. Recently, Tomar et al. [2020] and Vaswani et al. [2021] propose general policy optimization algorithms based on mirror descent that are similar to ours. However, both of them treat policy parameterization as a black box and neither provides a convergence rate analysis.

Other related work. The capability of policy gradient methods to do exploration in MDPs is also studied in Cai et al. [2020], Agarwal et al. [2020], Shani et al. [2020b], Zanette et al. [2021]. Grudzien et al. [2022] proposes an abstract framework called mirror learning for both tabular and continuous-space MDPs that includes mirror descent as a special case. It provides an asymptotic convergence analysis but does not consider any function approximation setting. Finally, for optimization in functional space, Chu et al. [2019] provides a framework setup that unifies variational inference and reinforcement learning. More recently, Aubin-Frankowski et al. [2022] studies mirror descent in general functional space and provides a rigorous convergence rate analysis. However, it only focuses on the primal space.

B Legendre Function and Relaxations

Let $\mathcal{X}$ be a normed vector space, possibly of infinite dimension, and $\Phi : \mathcal{X} \to \mathbb{R} \cup \{+\infty\}$ a proper, closed convex function with $\operatorname{dom}\Phi = \{x \in \mathcal{X} \mid \Phi(x) < +\infty\}$.

Definition B.1. The function Φ is of *Legendre type* if

(a) The interior of $\operatorname{dom}\Phi$, denoted by $\mathcal{D}$, is nonempty;

(b) Φ is differentiable and strictly convex on $\mathcal{D}$;

(c) For any sequence $\{x_n\} \subset \mathcal{D}$ which converges to a boundary point of $\mathcal{D}$, it holds that $\lim_{n\to\infty} \|\nabla\Phi(x_n)\| = \infty$.

Let $\mathcal{X}^*$ be the dual vector space of $\mathcal{X}$. The (Legendre) conjugate of Φ is defined as follows: for any $x^\star \in \mathcal{X}^\star$,

$$\Phi^*(x^*) = \sup_{x\in\operatorname{dom}\Phi} \{\langle x, x^*\rangle - \Phi(x)\}. \tag{28}$$

Similarly, $\operatorname{dom}\Phi^* = \{x^* \in \mathcal{X}^* \mid \Phi^*(x^*) < +\infty\}$ and $\mathcal{D}^* = \operatorname{int}(\operatorname{dom}\Phi^*)$. If Φ is of Legendre type, then its gradient $\nabla\Phi$ is one-to-one from $\mathcal{D}$ to $\mathcal{D}^*$ and $\nabla\Phi^* = (\nabla\Phi)^{-1}$; in other words, for any $x \in \mathcal{D}$ and $x^* \in \mathcal{D}^*$,

$$\nabla\Phi^*(\nabla\Phi(x)) = x, \qquad \nabla\Phi(\nabla\Phi^*(x^*)) = x^*. \tag{29}$$

See Rockafellar [1970, Theorem 26.5] for further details.

However, if Φ is not of Legendre type, then (29) may not hold. In particular, this is the case if the $\operatorname{dom}\Phi$ is the simplex $\Delta = \{x \in \mathbb{R}^n_+ \mid \sum_i x_i = 1\}$, which has an empty interior. In fact, such functions are not even differentiable. To see this, let $\Phi(x) = \phi(x) + \delta(x|\Delta)$ where ϕ is convex and differentiable over $\mathbb{R}^n$, and $\delta(\cdot|\Delta)$ is the indicator function of Δ, i.e., $\delta(x|\Delta) = 0$ if $x \in \Delta$ and $+\infty$ otherwise. Then Φ is not a differentiable function. However, it is subdifferentiable with subdifferential

$$\partial\Phi(x) = \{\nabla\phi(x) + c\mathbf{1} \mid c \in \mathbb{R}\}, \tag{30}$$

where $\mathbf{1} = [1 \quad \dots \quad 1]^\top$. Given the importance of simplex in studying MDPs, we present the following relaxation of (29), which is crucial for our main results.

Lemma B.2. *Suppose $\Phi(x) = \phi(x) + \delta(x|\mathcal{L})$ where ϕ is a convex function of Legendre type and $\mathcal{L}$ is an affine subspace. Assume that $\operatorname{int}(\operatorname{dom}\phi) \cap \mathcal{L} \neq \emptyset$. Then we have*

$$\nabla\Phi^*(\nabla\Phi(x)) = x, \qquad \forall\, x \in \operatorname{int}(\operatorname{dom}\phi) \cap \mathcal{L}.$$

And for any $x^* \in \operatorname{int}(\operatorname{dom}\Phi^*)$ and any $x, y \in \operatorname{dom}\Phi$,

$$\big\langle \nabla\Phi(\nabla\Phi^*(x^*)), x - y\big\rangle = \langle x^*, x - y\rangle,$$

where $\nabla\Phi(x)$ denotes any subgradient in $\partial\Phi(x)$.

Proof. Let $\mathcal{L} = x_0 + \mathcal{V}$ where $\mathcal{V}$ is a subspace, and denote $\mathcal{V}^\perp$ its orthogonal complement. First, it is commonly known that the subdifferential of an indicator function is a normal cone [Bertsekas, 2009]. Thus, we have $\partial\delta(x \mid \mathcal{L}) = \mathcal{N}_{\mathcal{L}}(x) \stackrel{\text{def}}{=} \{g' \mid \langle g', v + x_0 - x\rangle \le 0, \forall v \in \mathcal{V}\}$. That is, for any $g' \in \mathcal{N}_{\mathcal{L}}(x)$, we have $\langle g', v\rangle \le \langle g', x - x_0\rangle$ for any $v \in \mathcal{V}$. Since $\mathcal{V}$ is a subspace, for any $v \in \mathcal{V}$, we have $\alpha v \in \mathcal{V}$ for any $\alpha \in \mathbb{R}$. Therefore, we must have $\langle g', v\rangle = 0$ for any $v \in \mathcal{V}$ and $g' \in \mathcal{N}_{\mathcal{L}}(x)$. That is, we have $\mathcal{N}_{\mathcal{L}}(x) = \mathcal{V}^\perp$. (The reverse side is straightforward.)

Suppose $x \in \operatorname{int}(\operatorname{dom}\phi) \cap \mathcal{L}$. Then, according to the subdifferential calculus rule, we must have $\nabla\Phi(x) = \nabla\phi(x) + \xi$ for some $\xi \in \mathcal{N}_{\mathcal{L}}(x) = \mathcal{V}^\perp$. Let $x' \triangleq \nabla\Phi^*(\nabla\Phi(x))$. By definition of Φ^* and strict convexity of ϕ, we have

$$x' = \operatorname*{arg\,max}_{z\in\mathcal{L}\cap\operatorname{dom}\phi} \{\langle z, \nabla\phi(x) + \xi\rangle - \phi(z)\}.$$

The optimality condition of the above problem is

$$\nabla\phi(x) + \xi - \nabla\phi(x') \in \mathcal{N}_{\mathcal{L}\cap\operatorname{dom}\phi}(x') = \mathcal{V}^\perp + \mathcal{N}_{\operatorname{dom}\phi}(x'),$$

where the last equality above holds because $\mathcal{N}_{\mathcal{L}}(x) = \mathcal{V}^\perp$. Note that $x' = \nabla\Phi^*(\nabla\Phi(x))$ implies $\nabla\Phi(x) \in \partial\Phi(x') = \partial\phi(x') + \partial\delta(x'|\mathcal{L})$. As shown in Rockafellar [1967], $\partial\phi(x) = \emptyset$ for any $x \in \operatorname{bd}\operatorname{dom}\phi$ for a Legendre type function ϕ. Therefore, we must have $x' \in \operatorname{int}(\operatorname{dom}\phi)$, which then implies $\mathcal{N}_{\operatorname{dom}\phi}(x') = \{\mathbf{0}\}$. Thus, we have

$$\nabla\phi(x) + \xi - \nabla\phi(x') \in \mathcal{V}^\perp$$

Since $\xi \in \mathcal{V}^\perp$, we conclude that $\nabla\phi(x) - \nabla\phi(x') \in \mathcal{V}^\perp$. On the other hand, we have $x, x' \in \mathcal{L}$, which implies that $x - x' \in \mathcal{V}$. Therefore,

$$\langle \nabla\phi(x) - \nabla\phi(x'),\, x - x' \rangle = 0.$$

Since ϕ is strictly convex, we must have $x = x'$, thus proving $\nabla\Phi^*(\nabla\Phi(x)) = x$.

To prove the second statement, let $x' \triangleq \nabla\Phi^*(x^*)$, i.e.,

$$x' = \operatorname*{arg\,max}_{z \in \mathcal{L} \cap \operatorname{dom}\phi} \left\{ \langle z, x^* \rangle - \phi(z) \right\}.$$

By similar reasoning, we have $x' \in \operatorname{int}(\operatorname{dom}\phi)$. Thus, the optimality condition is $x^* - \nabla\phi(x') \in \mathcal{V}^\perp$, meaning $\nabla\phi(x') = x^* + \xi$ for some $\xi \in \mathcal{V}^\perp$. Meanwhile,

$$\nabla\Phi(\nabla\Phi^*(x^*)) = \nabla\Phi(x') = \nabla\phi(x') + \xi' = x^* + \xi + \xi',$$

where $\xi' \in \mathcal{V}^\perp$. Since $\xi, \xi' \in \mathcal{V}^\perp$ and $x - y \in \mathcal{V}$, we have

$$\langle \nabla\Phi(\nabla\Phi^*(x^*)), x - y \rangle = \langle x^*, x - y \rangle.$$

This finishes the proof. □

Notice that if $\operatorname{dom}\phi = \mathbb{R}^n_+$ and $\mathcal{L} = \{x \in \mathbb{R}^n | \mathbf{1}^T x = 1\}$, then $\operatorname{dom}\Phi = \operatorname{dom}\phi \cap \mathcal{L} = \Delta$. This is how we will invoke Lemma B.2 with ϕ being the negative entropy function. We call $\Phi(x)$ defined in Lemma B.2 as *relaxed Legendre-type function*.

Furthermore, we have the following corollary so that the Bregman divergence in Eq. (5) is also well-defined for the relaxed Legendre-type function.

Corollary B.3. *In the setting of Lemma B.2, for any $x, y, z \in \operatorname{dom}\Phi$ and $g \in \partial\Phi(z)$, we have*

$$\langle g, x - y \rangle = \langle \nabla\phi(z), x - y \rangle,$$

which makes expression $\langle \nabla\Phi(z), x - y \rangle$ well-defined for $x, y \in \operatorname{dom}\Phi$.

Proof. Again let $\mathcal{L} = x_0 + \mathcal{V}$. As shown in the proof of Lemma B.2, we have $\partial\Phi(z) = \nabla\phi(z) + \partial\delta(z \mid \mathcal{L})$ and $\partial\delta(z \mid \mathcal{L}) = \mathcal{V}^\perp$. Then, since $\operatorname{dom}\Phi \subseteq \mathcal{L}$, we have $x - y \in \mathcal{V}$, which means to have $\langle g', x - y \rangle = 0$ for any $g' \in \partial\delta(z \mid \mathcal{L})$. Therefore, we have

$$\langle g, x - y \rangle = \langle \nabla\phi(z), x - y \rangle + \langle g', x - y \rangle = \langle \nabla\phi(z), x - y \rangle.$$

□

The following corollary gives a dual relationship between Φ's and Φ^*'s Bregman divergences.

Corollary B.4. *In the setting of Lemma B.2, for $x^*, y^* \in \operatorname{int}(\operatorname{dom}\Phi^*)$, we have*

$$D_{\Phi^*}(x^*, y^*) = D_\Phi(\nabla\Phi^*(y^*), \nabla\Phi^*(x^*)).$$

Proof. By definition of the Bregman divergence in Eq. (5), we have

$$\begin{aligned}
& D_{\Phi^*}(x^*, y^*) - D_\Phi(\nabla\Phi^*(y^*), \nabla\Phi^*(x^*)) \\
=& \Phi^*(x^*) - \Phi^*(y^*) - \langle \nabla\Phi^*(y^*), x^* - y^* \rangle \\
& \quad - [\Phi(\nabla\Phi^*(y^*)) - \Phi(\nabla\Phi^*(x^*)) - \langle \nabla\Phi(\nabla\Phi^*(x^*)), \nabla\Phi^*(y^*) - \nabla\Phi^*(x^*) \rangle] \\
=& [\Phi^*(x^*) + \Phi(\nabla\Phi^*(x^*))] - [\Phi^*(y^*) + \Phi(\nabla\Phi^*(y^*))] \\
& \quad - \langle \nabla\Phi^*(y^*), x^* - y^* \rangle + \langle x^*, \nabla\Phi^*(y^*) - \nabla\Phi^*(x^*) \rangle \\
& \hfill \text{(By Lemma B.2 and } \nabla\Phi^*(y^*), \nabla\Phi^*(x^*) \in \operatorname{dom}\Phi.) \\
=& \langle x^*, \nabla\Phi^*(x^*) \rangle - \langle y^*, \nabla\Phi^*(y^*) \rangle - \langle \nabla\Phi^*(y^*), x^* - y^* \rangle + \langle x^*, \nabla\Phi^*(y^*) - \nabla\Phi^*(x^*) \rangle \\
& \hfill \text{(By Bertsekas [2009, Proposition 5.4.3].)} \\
=& \langle \nabla\Phi^*(x^*), x^* - x^* \rangle + \langle \nabla\Phi^*(y^*), -y^* + y^* - x^* + x^* \rangle \\
=& 0.
\end{aligned}$$

□

C Analysis of DAPO-L_2

The analysis of DAPO-KL requires a slightly modified assumption (A1') compared with DAPO-KL.

(A1') Under the same setting as (A1), we instead have

$$\mathbb{E}_{s\sim d_\rho^{(k)}}\Big[\big\|f_s^{(k+1)} - \big(\pi_s^{(k)} + \eta_k \widehat{Q}_s^{(k)}\big)\big\|_2^2\Big] \le 2\eta_k^2 \epsilon_{\mathrm{actor}}.$$

The scaling coefficient η_k^2 is consistent with Assumption (A1') in Alfano et al. [2023] as they assume the L_2-error for approximating Meanwhile, notice that in (A1) at Eq. (26), the upper bound of the approximation error is $\eta_k \epsilon_{\mathrm{actor}}$ and is different from (A1'). This is the result of considering the growth rate of the approximation error in η_k for different Bregman divergences. Specifically, if we keep $f^{(k+1)}$, $f^{(k)}$ and $\widehat{Q}^{(k)}$ fixed, then the L_2-error in (A1') satisfies

$$\mathbb{E}_{s\sim d_\rho^{(k)}}\Big[\big\|f_s^{(k+1)} - \big(\pi_s^{(k)} + \eta_k \widehat{Q}_s^{(k)}\big)\big\|_2^2\Big] \propto \eta_k^2.$$

However, the KL-divergence in (A1) satisfies

$$\mathbb{E}_{s\sim d_\rho^{(k)}}\Big[D_{\mathrm{KL}}\Big(\pi_s^{(k+1)} \Big\| \pi_s^{(k)} \exp\big(-\eta_k \widehat{Q}_s^{(k)}\big)/Z_s^{(k)}\Big)\Big] \propto \eta_k.$$

Then, we have the following theorem.

Theorem C.1 (Linear Convergence of DAPO-L_2)**.** *Consider Algorithm 1 with initial policy $\pi^{(0)}$, initial distribution $\rho \in \Delta(\mathcal{S})$ and Φ being the squared L_2-norm. Suppose Assumptions (A1') and (A2) hold and the step sizes satisfy $\eta_0 > 1$ and $\eta_{k+1} \ge (\vartheta_\rho/(\vartheta_\rho - 1))\,\eta_k$ for all $k \ge 0$. Then, for any comparator policy $\pi^\star$, it holds that*

$$V_\rho^{(K)} - V_\rho^\star \le \left(1 - \frac{1}{\vartheta_\rho}\right)^K \left(V_\rho^{(0)} - V_\rho^\star + \frac{D_0^\star/(\vartheta_\rho - 1)}{(1-\gamma)\eta_0}\right) + \frac{\vartheta_\rho^2\sqrt{2\epsilon_{\mathrm{actor}}} + 2\vartheta_\rho \epsilon_{\mathrm{critic}}}{1-\gamma}.$$

where $D_0^\star = \mathbb{E}_{s\sim d_\rho^\star}\big[D_\Phi(\pi_s^\star, \pi_s^{(0)})\big]$.

The proof of Theorem C.1 is given in Appendix D. It retains the convergence rate of Alfano et al. [2023] albeit with some different techniques. This is expected since in the L_2 case, DAPO-L_2 is the same as AMPO.

D Convergence Analysis of DAPO

The analysis starts by proving an approximate version of the Pythagorean theorem, which controls the error in three-point identity by the corresponding Bregman divergence and will serve as the key tool of our analysis.

D.1 Approximate Pythagorean Theorem

We begin with a general upper bound and then, we will derive its extensions under specific choices of mirror maps.

Lemma D.1 (Approximate Pythagorean Theorem)**.** *Let $\Phi : \mathcal{C} \mapsto \mathbb{R}$ be a proper closed convex mirror map, $\mathcal{D} \subseteq \mathcal{C}$ be a closed convex set and $v, c \in \mathcal{C}$ be two points. Suppose $u^\star = \arg\min_{u\in\mathcal{D}} D_\Phi(u, v)$. Then, for any $u \in \mathcal{D}$, we have*

$$D_\Phi(u, u^\star) + D_\Phi(u^\star, c) - D_\Phi(u, c) \le \langle \nabla\Phi(v) - \nabla\Phi(c), u^\star - u\rangle .$$

Proof. Using the definition of Bregman divergence in Eq. (5), we have

$$\begin{aligned} D_\Phi(u, u^\star) + D_\Phi(u^\star, c) - D_\Phi(u, c) &= \langle \nabla\Phi(u^\star), u - u^\star\rangle - \langle \nabla\Phi(c), u^\star - c\rangle + \langle \nabla\Phi(c), u - c\rangle \\ &= \langle \nabla\Phi(u^\star) - \nabla\Phi(c), u^\star - u\rangle \qquad (31) \\ &= \underbrace{\langle \nabla\Phi(u^\star) - \nabla\Phi(v), u^\star - u\rangle}_{\le 0 \text{ by Lemma 4.1 in Bubeck et al. [2012]}} + \langle \nabla\Phi(v) - \nabla\Phi(c), u^\star - u\rangle \\ &\le \langle \nabla\Phi(v) - \nabla\Phi(c), u^\star - u\rangle . \end{aligned}$$

□

D.1.1 Extension under Squared L_2-Norm

Lemma D.2. *Under the condition of Lemma D.1, if we take Φ to be the squared L_2-norm (see Example 2.1) and $\mathcal{D} = \Delta(\mathcal{A})$, then for any $u \in \mathcal{D}$, we have*

$$D_\Phi(u, u^\star) + D_\Phi(u^\star, c) - D_\Phi(u, c) \le \sqrt{2D_\Phi(v, c)}.$$

Proof. By Lemma D.1, we only need to bound $\langle \nabla\Phi(v) - \nabla\Phi(c), u^\star - u\rangle$. Then, since $\nabla\Phi(x) = x$, we have

$$\langle \nabla\Phi(v) - \nabla\Phi(c), u^\star - u\rangle \le \|v - c\|_2 \|u^\star - u\|_2 \le \sqrt{2D_\Phi(v, c)},$$

where $\|u^\star - u\|_2 \le 1$ since $u^\star, u \in \Delta(\mathcal{A})$. □

D.1.2 Extension under Negative Entropy

Lemma D.3. *Under the condition of Lemma D.1, if we take Φ to be the negative entropy restricted on $\Delta(\mathcal{A})$ (see Example 2.3) and assume $\mathcal{C} = \mathcal{D} = \Delta(\mathcal{A})$, then for any $u \in \mathcal{D}$, we have*

$$D_\Phi(u, u^\star) + D_\Phi(u^\star, c) - D_\Phi(u, c) \le \left(1 + \left\|\frac{u}{v}\right\|_\infty\right)\left(D_\Phi(v, c) + \sqrt{2D_\Phi(v, c)}\right).$$

Proof. By Lemma D.1, we only need to bound $\langle \nabla\Phi(v) - \nabla\Phi(c), u^\star - u\rangle$. Then, since $\mathcal{C} = \mathcal{D} = \Delta(\mathcal{A})$, we have $v = u^\star$. Therefore, we have

$$\begin{aligned}
\langle \nabla\Phi(v) - \nabla\Phi(c), u^\star - u\rangle &= \langle \nabla\Phi(v) - \nabla\Phi(c), v - u\rangle \\
&= \left\langle \log\frac{v}{c}, v - u\right\rangle \qquad \text{(Since } \Phi \text{ is the negative Shannon entropy)} \\
&= D_{\mathrm{KL}}(v\|c) - \left\langle \log\frac{v}{c}, u\right\rangle \\
&\le D_{\mathrm{KL}}(v\|c) + \left\|\frac{u}{v}\right\|_\infty \left\langle \left|\log\frac{v}{c}\right|, v\right\rangle \\
&\le D_{\mathrm{KL}}(v\|c) + \left\|\frac{u}{v}\right\|_\infty \left(D_{\mathrm{KL}}(v\|c) + \sqrt{2D_{\mathrm{KL}}(v\|c)}\right). \qquad \text{(By Lemma D.4)} \\
&\le \left(1 + \left\|\frac{u}{v}\right\|_\infty\right)\left(D_\Phi(v, c) + \sqrt{2D_\Phi(v, c)}\right)
\end{aligned}$$

□

Lemma D.4. *For any distributions $p, q \in \Delta(\mathcal{A})$ such that p is absolutely continuous with respect to q, we have $\left\langle \left|\log\frac{p}{q}\right|, p\right\rangle \le D_{\mathrm{KL}}(p\|q) + \sqrt{2D_{\mathrm{KL}}(p\|q)}$.*

Proof. Without loss of generality, assume $\mathrm{supp}(p) = \mathcal{A}$. Now, we define $\mathcal{A}^+ = \{a \in \mathcal{A} \mid p_a \ge q_a\}$ and $\mathcal{A}^- = \{a \in \mathcal{A} \mid p_a < q_a\}$. Then, when p, q are discrete distributions, we have

$$\begin{aligned}
\left\langle \left|\log\frac{p}{q}\right|, p\right\rangle &= \sum_{a\in\mathcal{A}^+} p_a \log\frac{p_a}{q_a} + \sum_{a\in\mathcal{A}^-} p_a \log\frac{q_a}{p_a} \\
&= \sum_{a\in\mathcal{A}^+} p_a \log\frac{p_a}{q_a} - \sum_{a\in\mathcal{A}^-} p_a \log\frac{q_a}{p_a} + \sum_{a\in\mathcal{A}^-} p_a \log\frac{q_a}{p_a} + \sum_{a\in\mathcal{A}^-} p_a \log\frac{q_a}{p_a} \\
&= D_{\mathrm{KL}}(p\|q) + 2\sum_{a\in\mathcal{A}^-} p_a \log\frac{q_a}{p_a} \\
&\le D_{\mathrm{KL}}(p\|q) + 2\sum_{a\in\mathcal{A}^-} p_a \left(\frac{q_a}{p_a} - 1\right) \qquad \text{(Since } \log x \le x - 1 \text{ for any } x > 0.\text{)} \\
&= D_{\mathrm{KL}}(p\|q) + 2\sum_{a\in\mathcal{A}^-} (q_a - p_a) \\
&= D_{\mathrm{KL}}(p\|q) + 2\|q - p\|_{\mathrm{TV}} \qquad \text{(By definition of total variation distance.)} \\
&\le D_{\mathrm{KL}}(p\|q) + \sqrt{2D_{\mathrm{KL}}(p\|q)}. \qquad \text{(By Pinsker's inequality.)}
\end{aligned}$$

□

D.2 Proof of Theorem C.1 and Theorem 4.1

We first recall the Assumption (A1), (A1'), (A2) and (A3) listed in Section 4. Here, we prove Theorem C.1 and Theorem 4.1 in a slightly more general version in which the training data distributions can be different from $d_\rho^{(k)}$ as long as they satisfy the assumptions. This slight extension makes our result applicable to the offline training setting where $\nu^{(k)} \in \Delta(\mathcal{S})$ is the reply buffer distribution at k-th iteration. Taking $\nu^{(k)} = d_\rho^{(k)}$ recovers the online training setting.

(A1) With initial distribution $\rho \in \Delta(\mathcal{S})$ and reply buffer distribution $\nu^{(k)} \in \Delta(\mathcal{S})$, there exist constants $\epsilon_{\text{critic}}, \epsilon_{\text{actor}} > 0$ such that for any k, it holds

$$\mathbb{E}_{s\sim\nu^{(k)}}\left[\left\|\widehat{Q}_s^{(k)} - Q_s^{(k)}\right\|_\infty\right] \le \epsilon_{\text{critic}},$$
$$\mathbb{E}_{s\sim\nu^{(k)}}\left[D_{\text{KL}}\left(\pi_s^{(k+1)} \middle\| \frac{\pi_s^{(k)}\exp\left(-\eta_k \widehat{Q}_s^{(k)}\right)}{Z_s^{(k)}}\right)\right] \le \eta_k \epsilon_{\text{actor}},$$

(A1') Under the same setting as (A1), we instead have

$$\mathbb{E}_{s\sim\nu^{(k)}}\left[\frac{1}{2}\left\|f_s^{(k+1)} - \left(\pi_s^{(k)} + \eta_k \widehat{Q}_s^{(k)}\right)\right\|_2^2\right] \le \eta_k^2 \epsilon_{\text{actor}}.$$

(A2) With initial distribution ρ and replay buffer distribution $\nu^{(k)} \in \Delta(\mathcal{S})$, there exists constant $\vartheta_\rho \ge 1$ such that for any k, it holds

$$\max\left\{\left\|\frac{d_\rho^\star}{d_\rho^{(k+1)}}\right\|_\infty, \left\|\frac{d_\rho^{(k+1)}}{\nu^{(k)}}\right\|_\infty, \left\|\frac{d_{d_\rho^\star}^{(k+1)}}{\nu^{(k)}}\right\|_\infty, \left\|\frac{d_{d_\rho^\star}^{(k+1)}}{d_\rho^\star}\right\|_\infty\right\} \le \vartheta_\rho.$$

(A3) There exists constant $C_\rho > 0$ such that for any k, it holds

$$\max_{s\in\text{supp}(\nu^{(k)})}\left\{\left\|\frac{\pi_s^\star}{\pi_s^{(k+1)}}\right\|_\infty, \left\|\frac{\pi_s^{(k)}}{\pi_s^{(k+1)}}\right\|_\infty\right\} \le C_\rho.$$

To present the proof in an unified way, we further define $C_{\rho,s} = \max\left\{\left\|\frac{\pi_s^\star}{\pi_s^{(k+1)}}\right\|_\infty, \left\|\frac{\pi_s^{(k)}}{\pi_s^{(k+1)}}\right\|_\infty\right\}$ for some state $s \in \mathcal{S}$ and $\psi_s^\Phi : \mathbb{R}_+ \mapsto \mathbb{R}_+$ as

$$\psi_s^\Phi(x) = \begin{cases} \sqrt{2x}, & \text{if } \Phi \text{ is squared } L_2\text{-norm},\\ (1 + C_{\rho,s})\left(x + \sqrt{2x}\right), & \text{if } \Phi \text{ is the negative entropy on } \Delta(\mathcal{A}). \end{cases} \tag{32}$$

Then, applying Lemma D.2 and D.3 to Algorithm 1 will result the following key lemma.

Lemma D.5. *Consider running Algorithm 1. Then, for policy $\pi = \pi^{(k)}$ or $\pi = \pi^\star$, for any $s \in \mathcal{S}$, if Φ is either squared L_2-norm or negative entropy on $\Delta(\mathcal{A})$, we have*

$$\begin{aligned}&\eta_k\left\langle \widehat{Q}_s^{(k)}, \pi_s^{(k+1)} - \pi_s\right\rangle + D_\Phi\left(\pi_s, \pi_s^{(k+1)}\right) + D_\Phi\left(\pi_s^{(k+1)}, \pi_s^{(k)}\right) - D_\Phi\left(\pi_s, \pi_s^{(k)}\right)\\ \le&\psi_s^\Phi\left(D_\Phi\left(\nabla\Phi^*(f_s^{(k+1)}), \nabla\Phi^*\left(\nabla\Phi(\pi_s^{(k)}) - \eta_k\widehat{Q}_s^{(k)}\right)\right)\right).\end{aligned}$$

Proof. Fix some $s \in \mathcal{S}$. Since line 5 of Algorithm 1 states that

$$\pi_s^{(k+1)} \in \underset{\pi_s' \in \Delta(\mathcal{A})}{\arg\min}\, D_\Phi\left(\pi_s', \nabla\Phi^*(f_s^{(k+1)})\right),$$

Then, We can apply Lemma D.2 or D.3 with $\mathcal{D} = \Delta(\mathcal{A})$, $u = \pi_s$, $u^\star = \pi_s^{(k+1)}$, $v = \nabla\Phi^*(f_s^{(k+1)})$ and $c = \nabla\Phi^*\left(\nabla\Phi(\pi_s^{(k)}) - \eta_k\widehat{Q}_s^{(k)}\right)$, which gives us

$$D_\Phi\left(\pi_s^{(k+1)}, \nabla\Phi^*\left(\nabla\Phi(\pi_s^{(k)}) - \eta_k\widehat{Q}_s^{(k)}\right)\right) - D_\Phi\left(\pi_s, \nabla\Phi^*\left(\nabla\Phi(\pi_s^{(k)}) - \eta_k\widehat{Q}_s^{(k)}\right)\right)$$

$$+ D_\Phi\left(\pi_s, \pi_s^{(k+1)}\right) \leq \psi_s^\Phi\left(D_\Phi\left(\nabla\Phi^*(f_s^{(k+1)}), \nabla\Phi^*\left(\nabla\Phi(\pi_s^{(k)}) - \eta_k \widehat{Q}_s^{(k)}\right)\right)\right).$$

By using the identity in Eq. (31), for the left-hand side of the above inequality, we have

$$\begin{aligned}
\text{LHS} =& \left\langle \nabla\Phi(\pi_s^{(k+1)}) - \nabla\Phi\left(\nabla\Phi^*\left(\nabla\Phi(\pi_s^{(k)}) - \eta_k \widehat{Q}_s^{(k)}\right)\right), \pi_s^{(k+1)} - \pi_s \right\rangle \\
=& \left\langle \nabla\Phi(\pi_s^{(k+1)}) - \left(\nabla\Phi(\pi_s^{(k)}) - \eta_k \widehat{Q}_s^{(k)}\right), \pi_s^{(k+1)} - \pi_s \right\rangle \qquad \text{(By Lemma B.2.)} \\
=& \eta_k \left\langle \widehat{Q}_s^{(k)}, \pi_s^{(k+1)} - \pi_s \right\rangle + \left\langle \nabla\Phi(\pi_s^{(k+1)}) - \nabla\Phi(\pi_s^{(k)}), \pi_s^{(k+1)} - \pi_s \right\rangle \\
=& \eta_k \left\langle \widehat{Q}_s^{(k)}, \pi_s^{(k+1)} - \pi_s \right\rangle + D_\Phi\left(\pi_s, \pi_s^{(k+1)}\right) + D_\Phi\left(\pi_s^{(k+1)}, \pi_s^{(k)}\right) - D_\Phi\left(\pi_s, \pi_s^{(k)}\right)
\end{aligned}$$

(By using the identity in Eq. (31) again on the second term above.)

The proof is then complete by plugging this inequality back. □

Notice that the conditions in Assumption (A1) and (A1') can unifiedly written as

$$\mathbb{E}_{s\sim\nu^{(k)}}\left[D_\Phi\left(\nabla\Phi^*(f_s^{(k+1)}), \nabla\Phi^*\left(\nabla\Phi(\pi_s^{(k)}) - \eta_k \widehat{Q}_s^{(k)}\right)\right)\right] \leq \eta_k^{\omega^\Phi} \epsilon_{\text{actor}},$$

where we define ω^Φ as

$$\omega^\Phi = \begin{cases} 2, & \text{if } \Phi \text{ is squared } L_2\text{-norm}, \\ 1, & \text{if } \Phi \text{ is the negative entropy on } \Delta(\mathcal{A}). \end{cases}$$

Therefore, we can summarize both Theorem C.1 and 4.1 into the following theorem and present its proof.

Theorem D.6. *Consider Algorithm 1 with initial policy $\pi^{(0)}$, initial distribution $\rho \in \Delta(\mathcal{S})$ and Φ being either the squared L_2-norm or negative entropy on $\Delta(\mathcal{A})$. Let Assumption (A1), (A1'), (A2), (A3) hold and suppose the learning rates satisfy $\eta_0 \geq 1$ and $\eta_{k+1} \geq \frac{\vartheta_\rho}{\vartheta_\rho - 1}\eta_k$ for any $k \in [K]$. Then, for any comparator policy $\pi^\star$, with $D_0^\star = \mathbb{E}_{s\sim d_\rho^\star}\left[D_\Phi(\pi_s^\star, \pi_s^{(0)})\right]$, it holds that*

$$V_\rho^{(K)} - V_\rho^\star \leq \left(1 - \frac{1}{\vartheta_\rho}\right)^K \left(V_\rho^{(0)} - V_\rho^\star + \frac{D_0^\star/(\vartheta_\rho - 1)}{(1-\gamma)\eta_0}\right) + \frac{\vartheta_\rho^2 \psi^\Phi(\epsilon_{\text{actor}}) + 2\vartheta_\rho \epsilon_{\text{critic}}}{1-\gamma},$$

where we define $\psi^\Phi : \mathbb{R}_+ \mapsto \mathbb{R}_+$ as

$$\psi^\Phi(x) = \begin{cases} \sqrt{2x}, & \text{if } \Phi \text{ is } L_2\text{-norm square}, \\ (1 + C_\rho)\left(x + \sqrt{2x}\right), & \text{if } \Phi \text{ is the negative entropy}. \end{cases} \tag{33}$$

Proof. ***Step 1.*** Fix some $s \in \mathcal{S}$ and $k < K$. First, by Lemma D.5 with $\pi_s = \pi_s^{(k)}$, we have

$$\begin{aligned}
&\eta_k \left\langle \widehat{Q}_s^{(k)}, \pi_s^{(k+1)} - \pi_s^{(k)} \right\rangle + D_\Phi\left(\pi_s^{(k)}, \pi_s^{(k+1)}\right) \\
\leq& \psi_s^\Phi\left(D_\Phi\left(\nabla\Phi^*(f_s^{(k+1)}), \nabla\Phi^*\left(\nabla\Phi(\pi_s^{(k)}) - \eta_k \widehat{Q}_s^{(k)}\right)\right)\right),
\end{aligned}$$

where we dropped the term $D_\Phi\left(\pi_s^{(k+1)}, \pi_s^{(k)}\right)$ since it is always non-negative.

Since $D_\Phi\left(\pi_s^{(k)}, \pi_s^{(k+1)}\right) \geq 0$ as a Bregman divergence, we have

$$\Delta_s^{(k)} \stackrel{\text{def}}{=} \eta_k \left\langle \widehat{Q}_s^{(k)}, \pi_s^{(k+1)} - \pi_s^{(k)} \right\rangle - \psi_s^\Phi\left(D_\Phi\left(\nabla\Phi^*(f_s^{(k+1)}), \nabla\Phi^*\left(\nabla\Phi(\pi_s^{(k)}) - \eta_k \widehat{Q}_s^{(k)}\right)\right)\right) \leq 0. \tag{34}$$

Then, by using Lemma D.5 with $\pi_s = \pi_s^\star$, the comparator policy, and similarly dropping $D_\Phi\left(\pi_s^{(k+1)}, \pi_s^{(k)}\right)$, we have

$$\eta_k \left\langle \widehat{Q}_s^{(k)}, \pi_s^{(k+1)} - \pi_s^\star \right\rangle + D_\Phi\left(\pi_s^\star, \pi_s^{(k+1)}\right) - D_\Phi\left(\pi_s^\star, \pi_s^{(k)}\right)$$

$$\leq \psi_s^{\Phi}\left(D_{\Phi}\left(\nabla\Phi^*(f_s^{(k+1)}), \nabla\Phi^*\left(\nabla\Phi(\pi_s^{(k)}) - \eta_k \widehat{Q}_s^{(k)}\right)\right)\right).$$

By adding and subtracting $\eta_k \left\langle \widehat{Q}_s^{(k)}, \pi_s^{(k)} \right\rangle$ together with some rearrangement, we have

$$\eta_k \left\langle \widehat{Q}_s^{(k)}, \pi_s^{(k+1)} - \pi_s^{(k)} \right\rangle - \psi_s^{\Phi}\left(D_{\Phi}\left(\nabla\Phi^*(f_s^{(k+1)}), \nabla\Phi^*\left(\nabla\Phi(\pi_s^{(k)}) - \eta_k \widehat{Q}_s^{(k)}\right)\right)\right)$$
$$+ \eta_k \left\langle \widehat{Q}_s^{(k)}, \pi_s^{(k)} - \pi_s^{\star} \right\rangle \leq D_{\Phi}\left(\pi_s^{\star}, \pi_s^{(k)}\right) - D_{\Phi}\left(\pi_s^{\star}, \pi_s^{(k+1)}\right).$$

Taking expectation on both sides with respect to distribution $d_\rho^\star$, we have

$$\mathbb{E}_{s\sim d_\rho^\star}\left[\eta_k \left\langle \widehat{Q}_s^{(k)}, \pi_s^{(k+1)} - \pi_s^{(k)} \right\rangle - \psi_s^{\Phi}\left(D_{\Phi}\left(\nabla\Phi^*(f_s^{(k+1)}), \nabla\Phi^*\left(\nabla\Phi(\pi_s^{(k)}) - \eta_k \widehat{Q}_s^{(k)}\right)\right)\right)\right]$$
$$+ \eta_k \mathbb{E}_{s\sim d_\rho^\star}\left[\left\langle \widehat{Q}_s^{(k)}, \pi_s^{(k)} - \pi_s^{\star} \right\rangle\right] \leq D_k^\star - D_{k+1}^\star, \tag{35}$$

where $D_k^\star = \mathbb{E}_{s\sim d_\rho^\star}\left[D_{\Phi}\left(\pi_s^{\star}, \pi_s^{(k)}\right)\right]$.

Then, for the first expectation above, we have

$$\begin{aligned}
&\mathbb{E}_{s\sim d_\rho^\star}\left[\Delta_s^{(k)}\right] \\
\geq &\left\|\frac{d_\rho^\star}{d_\rho^{(k+1)}}\right\|_\infty \mathbb{E}_{s\sim d_\rho^{(k+1)}}\left[\Delta_s^{(k)}\right] && \text{(By Eq. (34).)} \\
\geq &\vartheta_\rho \mathbb{E}_{s\sim d_\rho^{(k+1)}}\left[\Delta_s^{(k)}\right] && \text{(By Assumption (A2).)} \\
= &\eta_k \vartheta_\rho \mathbb{E}_{s\sim d_\rho^{(k+1)}}\left[\left\langle \widehat{Q}_s^{(k)}, \pi_s^{(k+1)} - \pi_s^{(k)} \right\rangle\right] \\
&- \vartheta_\rho^2 \mathbb{E}_{s\sim \nu^{(k)}}\left[\psi_s^{\Phi}\left(D_{\Phi}\left(\nabla\Phi^*(f_s^{(k+1)}), \nabla\Phi^*\left(\nabla\Phi(\pi_s^{(k)}) - \eta_k \widehat{Q}_s^{(k)}\right)\right)\right)\right] \\
& && \text{(By Assumption (A2).)} \\
\geq &\eta_k \vartheta_\rho \mathbb{E}_{s\sim d_\rho^{(k+1)}}\left[\left\langle Q_s^{(k)}, \pi_s^{(k+1)} - \pi_s^{(k)} \right\rangle\right] + \eta_k \vartheta_\rho \mathbb{E}_{s\sim d_\rho^{(k+1)}}\left[\left\langle \widehat{Q}_s^{(k)} - Q_s^{(k)}, \pi_s^{(k+1)} - \pi_s^{(k)} \right\rangle\right] \\
&- \vartheta_\rho^2 \psi^{\Phi}\left(\mathbb{E}_{s\sim \nu^{(k)}}\left[D_{\Phi}\left(\nabla\Phi^*(f_s^{(k+1)}), \nabla\Phi^*\left(\nabla\Phi(\pi_s^{(k)}) - \eta_k \widehat{Q}_s^{(k)}\right)\right)\right]\right) \\
& && \text{(By Assumption (A3), Jensen's inequality and concavity of } \psi_s^{\Phi}.) \\
\overset{\text{(i)}}{\geq} &\eta_k \vartheta_\rho \mathbb{E}_{s\sim d_\rho^{(k+1)}}\left[\left\langle Q_s^{(k)}, \pi_s^{(k+1)} - \pi_s^{(k)} \right\rangle\right] - \eta_k \vartheta_\rho \epsilon_{\text{critic}} - \vartheta_\rho^2 \psi^{\Phi}\left(\eta_k^{\omega^{\Phi}} \epsilon_{\text{actor}}\right) \\
& && \text{(By Assumption (A1), (A1') and monotonicity of } \psi^{\Phi}.) \\
= &(1-\gamma)\,\eta_k \vartheta_\rho \left(V_\rho^{(k+1)} - V_\rho^{(k)}\right) - \eta_k \vartheta_\rho \epsilon_{\text{critic}} - \vartheta_\rho^2 \psi^{\Phi}\left(\eta_k^{\omega^{\Phi}} \epsilon_{\text{actor}}\right) && \text{(By Lemma D.8.)}
\end{aligned}$$

The inequality (i) above holds also because by Hölder's inequality, we have

$$\left\langle \widehat{Q}_s^{(k)} - Q_s^{(k)}, \pi_s^{(k+1)} - \pi_s^{(k)} \right\rangle \leq \left\|\widehat{Q}_s^{(k)} - Q_s^{(k)}\right\|_\infty \left\|\pi_s^{(k+1)} - \pi_s^{(k)}\right\|_1 \leq \left\|\widehat{Q}_s^{(k)} - Q_s^{(k)}\right\|_\infty.$$

Similarly, for the second expectation in Eq. (35), we have

$$\begin{aligned}
&\eta_k \mathbb{E}_{s\sim d_\rho^\star}\left[\left\langle \widehat{Q}_s^{(k)}, \pi_s^{(k)} - \pi_s^{\star} \right\rangle\right] \\
= &\eta_k \mathbb{E}_{s\sim d_\rho^\star}\left[\left\langle Q_s^{(k)}, \pi_s^{(k)} - \pi_s^{\star} \right\rangle\right] + \eta_k \mathbb{E}_{s\sim d_\rho^\star}\left[\left\langle \widehat{Q}_s^{(k)} - Q_s^{(k)}, \pi_s^{(k)} - \pi_s^{\star} \right\rangle\right] \\
\geq &(1-\gamma)\eta_k \left(V_\rho^{(k)} - V_\rho^\star\right) - \eta_k \vartheta_\rho \epsilon_{\text{critic}}.
\end{aligned}$$

By plugging the results above back into Eq. (35) and defining $\delta_k \overset{\text{def}}{=} V_\rho^{(k)} - V_\rho^\star$, we have

$$\vartheta_\rho\left(\delta_{k+1} - \delta_k\right) + \delta_k \leq \frac{1}{(1-\gamma)\eta_k} D_k^\star - \frac{1}{(1-\gamma)\eta_k} D_{k+1}^\star + \frac{\vartheta_\rho^2 \psi^{\Phi}\left(\eta_k^{\omega^{\Phi}} \epsilon_{\text{actor}}\right)}{(1-\gamma)\eta_k} + \frac{2\vartheta_\rho \epsilon_{\text{critic}}}{1-\gamma}. \tag{36}$$

Step 2. Now, dividing both sides of Eq. (36) by ϑ_ρ together with some rearrangement, we can have

$$\delta_{k+1}+\frac{D^\star_{k+1}}{(1-\gamma)\eta_k\vartheta_\rho} \le \left(1-\frac{1}{\vartheta_\rho}\right)\left(\delta_k + \frac{D^\star_k}{(1-\gamma)\eta_k\left(\vartheta_\rho-1\right)}\right)+\frac{\vartheta_\rho\psi^\Phi\left(\eta_k^{\omega^\Phi}\epsilon_{\text{actor}}\right)}{(1-\gamma)\eta_k}+\frac{2\vartheta_\rho\epsilon_{\text{critic}}}{(1-\gamma)\vartheta_\rho}.$$

Since the learning rates satisfy $\eta_{k+1}(\vartheta_\rho-1)\ge\eta_k\vartheta_\rho$, we have

$$\begin{aligned}
&\delta_{k+1}+\frac{D^\star_{k+1}}{(1-\gamma)\eta_{k+1}(\vartheta_\rho-1)}\\
&\le\left(1-\frac{1}{\vartheta_\rho}\right)\left(\delta_k + \frac{D^\star_k}{(1-\gamma)\eta_k\left(\vartheta_\rho-1\right)}\right)+\frac{\vartheta_\rho\psi^\Phi\left(\eta_k^{\omega^\Phi}\epsilon_{\text{actor}}\right)}{(1-\gamma)\eta_k}+\frac{2\vartheta_\rho\epsilon_{\text{critic}}}{(1-\gamma)\vartheta_\rho}\\
&\le\left(1-\frac{1}{\vartheta_\rho}\right)\left(\delta_k + \frac{D^\star_k}{(1-\gamma)\eta_k\left(\vartheta_\rho-1\right)}\right)+\frac{\vartheta_\rho\psi^\Phi\left(\epsilon_{\text{actor}}\right)}{(1-\gamma)}+\frac{2\vartheta_\rho\epsilon_{\text{critic}}}{(1-\gamma)\vartheta_\rho},
\end{aligned}$$

where the second inequality above holds because we can straightforwardly verify that $\frac{\psi^\Phi(\eta_k^{\omega^\Phi}\epsilon_{\text{actor}})}{\eta_k}\le\psi^\Phi(\epsilon_{\text{actor}})$ for either choice of Φ. Then, applying the above relation recursively, we have

$$\begin{aligned}
\delta_K+\frac{D^\star_K}{(1-\gamma)\eta_K(\vartheta_\rho-1)} &\le \left(1-\frac{1}{\vartheta_\rho}\right)^K\left(\delta_0+\frac{D^\star_0}{(1-\gamma)\eta_0(\vartheta_\rho-1)}\right)\\
&\quad+\frac{\vartheta_\rho\psi^\Phi\left(\epsilon_{\text{actor}}\right)}{1-\gamma}\sum_{k=0}^{K-1}\left(1-\frac{1}{\vartheta_\rho}\right)^k+\frac{2\vartheta_\rho\epsilon_{\text{critic}}}{(1-\gamma)\vartheta_\rho}\sum_{k=0}^{K-1}\left(1-\frac{1}{\vartheta_\rho}\right)^k.
\end{aligned}\tag{37}$$

We can notice that

$$\sum_{k=0}^{K-1}\left(1-\frac{1}{\vartheta_\rho}\right)^k\le\frac{1}{1-\left(1-\frac{1}{\vartheta_\rho}\right)}=\vartheta_\rho.$$

Therefore, dropping the term with $D^\star_K$ in Eq. (37), we can finally have

$$V_\rho^{(K)}-V_\rho^\star\le\left(1-\frac{1}{\vartheta_\rho}\right)^K\left(V_\rho^{(0)}-V_\rho^\star+\frac{D_0^\star/(\vartheta_\rho-1)}{(1-\gamma)\eta_0}\right)+\frac{\vartheta_\rho^2\psi^\Phi(\epsilon_{\text{actor}})+2\vartheta_\rho\epsilon_{\text{critic}}}{1-\gamma}.$$

□

Then, Theorem C.1 and 4.1 are immediate consequences of Theorem D.6.

Theorem C.1 (Linear Convergence of DAPO-L_2)**.** *Consider Algorithm 1 with initial policy $\pi^{(0)}$, initial distribution $\rho\in\Delta(\mathcal{S})$ and Φ being the squared L_2-norm. Suppose Assumptions (A1') and (A2) hold and the step sizes satisfy $\eta_0>1$ and $\eta_{k+1}\ge\left(\vartheta_\rho/(\vartheta_\rho-1)\right)\eta_k$ for all $k\ge0$. Then, for any comparator policy $\pi^\star$, it holds that*

$$V_\rho^{(K)}-V_\rho^\star\le\left(1-\frac{1}{\vartheta_\rho}\right)^K\left(V_\rho^{(0)}-V_\rho^\star+\frac{D_0^\star/(\vartheta_\rho-1)}{(1-\gamma)\eta_0}\right)+\frac{\vartheta_\rho^2\sqrt{2\epsilon_{\text{actor}}}+2\vartheta_\rho\epsilon_{\text{critic}}}{1-\gamma}.$$

where $D_0^\star=\mathbb{E}_{s\sim d_\rho^\star}\left[D_\Phi(\pi_s^\star,\pi_s^{(0)})\right]$.

Proof. Apply $\psi^\Phi(x)=\sqrt{2x}$ to Theorem D.6. □

Theorem 4.1 (Linear Convergence of DAPO-KL)**.** *Consider Algorithm 1 with initial policy $\pi^{(0)}$, initial distribution $\rho\in\Delta(\mathcal{S})$ and Φ being the negative entropy restricted on $\Delta(\mathcal{A})$. Suppose Assumptions (A1), (A2) and (A3) hold and the step sizes satisfy $\eta_0>1$ and $\eta_{k+1}\ge\left(\vartheta_\rho/(\vartheta_\rho-1)\right)\eta_k$ for all $k\ge0$. Then, for any comparator policy $\pi^\star$, we have*

$$V_\rho^{(K)}-V_\rho^\star\le\left(1-\frac{1}{\vartheta_\rho}\right)^K\left(V_\rho^{(0)}-V_\rho^\star+\frac{D_0^\star/(\vartheta_\rho-1)}{(1-\gamma)\eta_0}\right)+\frac{\vartheta_\rho^2\psi(\epsilon_{\text{actor}})+2\vartheta_\rho\epsilon_{\text{critic}}}{1-\gamma},$$

where $\psi(x)=(1+C_\rho)\left(x+\sqrt{2x}\right)$ for $x\ge0$.

Proof. Apply $\psi^\Phi(x)=(1+C_\rho)\left(x+\sqrt{2x}\right)$ to Theorem D.6. □

D.3 Technical Lemmas

Lemma D.7. *If we take $\rho = \mathsf{Unif}(\mathcal{S})$, then for any k and any policy π, it holds that $\left\| \frac{d_\rho^\pi}{d_\rho^{(k)}} \right\|_\infty \leq \frac{|\mathcal{S}|}{1-\gamma}$.*

Proof. By definition of state-visitation distribution in Eq. (3), we can immediately get $d_{\rho,s}^{(k)} \geq (1-\gamma)\rho_s$ for any $s \in \mathcal{S}$ by truncating all terms with $t \geq 1$. Since $\rho = \mathsf{Unif}(\mathcal{S})$, we have $\rho_s = \frac{1}{|\mathcal{S}|}$ for any $s \in \mathcal{S}$. Thus, we have

$$\left\| \frac{d_\rho^\pi}{d_\rho^{(k)}} \right\|_\infty = \max_{s \in \mathcal{S}} \frac{d_{\rho,s}^\pi}{d_{\rho,s}^{(k)}} \leq \frac{1}{(1-\gamma)\rho_s} \leq \frac{|\mathcal{S}|}{1-\gamma}.$$

□

Lemma D.8 (Performance Difference Lemma). *For any two policies $\pi, \tilde{\pi} : \mathcal{S} \mapsto \Delta(\mathcal{A})$ and initial distribution $\rho \in \Delta(\mathcal{S})$, it holds that*

$$V_\rho^\pi - V_\rho^{\tilde{\pi}} = \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_\rho^\pi} \left[\left\langle Q_s^{\tilde{\pi}}, \pi_s - \tilde{\pi}_s \right\rangle \right] = \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_\rho^{\tilde{\pi}}} \left[\langle Q_s^\pi, \pi_s - \tilde{\pi}_s \rangle \right].$$

Proof. See Lemma 1 in Xiao [2022]. □

E Convergence Analysis of SAC

In this section, we prove a sublinear convergence rate for SAC under general function approximation by using our framework. It essentially adopts our proof techniques in Theorem D.6 to an entropy-regularized objective.

We start by presenting a modified version of the performance difference lemma under entropy-regularized reinforcement learning.

Lemma 4.3 (Modified Performance Difference Lemma). *For any two policies $\pi, \tilde{\pi} : \mathcal{S} \mapsto \Delta(\mathcal{A})$, initial distribution $\rho \in \Delta(\mathcal{S})$ and regularization strength $\tau > 0$, it holds that*

$$\begin{aligned} V_{\tau,\rho}^\pi - V_{\tau,\rho}^{\tilde{\pi}} &= \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_\rho^\pi} \left[\left\langle Q_{\tau,s}^{\tilde{\pi}}, \pi_s - \tilde{\pi}_s \right\rangle + \tau D_{\mathrm{KL}}(\pi_s \| \tilde{\pi}_s) \right] \\ &= \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_\rho^{\tilde{\pi}}} \left[\left\langle Q_{\tau,s}^\pi, \pi_s - \tilde{\pi}_s \right\rangle - \tau D_{\mathrm{KL}}(\tilde{\pi}_s \| \pi_s) \right]. \end{aligned}$$

Proof. By definition of the value function, we have

$$\begin{aligned} V_{\tau,\rho}^\pi - V_{\tau,\rho}^{\tilde{\pi}} =& \mathbb{E}_{a_t \sim \pi_{s_t}} \left[\sum_{t=0}^\infty \gamma^t \left(c(s_t, a_t) + \tau \log \pi(a_t \mid s_t) \right) \middle| s_0 \sim \rho \right] - V_{\tau,\rho}^{\tilde{\pi}} \\ =& \mathbb{E}_{a_t \sim \pi_{s_t}} \left[\sum_{t=0}^\infty \gamma^t \left(c(s_t, a_t) + \tau \log \pi(a_t \mid s_t) + \gamma V_\tau^{\tilde{\pi}}(s_{t+1}) - V_\tau^{\tilde{\pi}}(s_t) \right) \middle| s_0 \sim \rho \right] \\ =& \mathbb{E}_{a_t \sim \pi_{s_t}} \left[\sum_{t=0}^\infty \gamma^t \left(Q_\tau^{\tilde{\pi}}(s_t, a_t) - V_\tau^{\tilde{\pi}}(s_t) + \tau \log \frac{\pi(a_t \mid s_t)}{\tilde{\pi}(a_t \mid s_t)} \right) \middle| s_0 \sim \rho \right] \\ =& \mathbb{E}_{a_t \sim \pi_{s_t}} \left[\sum_{t=0}^\infty \gamma^t \left(A_\tau^{\tilde{\pi}}(s_t, a_t) + \tau \log \frac{\pi(a_t \mid s_t)}{\tilde{\pi}(a_t \mid s_t)} \right) \middle| s_0 \sim \rho \right] \\ & \text{(We define } A_\tau^\pi(s,a) = Q_\tau^\pi(s,a) - V_\tau^\pi(s).\text{)} \\ =& \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_\rho^\pi} \left[\left\langle Q_{\tau,s}^{\tilde{\pi}}, \pi_s - \tilde{\pi}_s \right\rangle + \tau D_{\mathrm{KL}}(\pi_s \| \tilde{\pi}_s) \right]. \end{aligned}$$

Then, by similarly expanding the term $V_{\tau,\rho}^{\tilde{\pi}}$, we can get

$$V_{\tau,\rho}^\pi - V_{\tau,\rho}^{\tilde{\pi}} = \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_\rho^{\tilde{\pi}}} \left[\left\langle Q_{\tau,s}^\pi, \pi_s - \tilde{\pi}_s \right\rangle - \tau D_{\mathrm{KL}}(\tilde{\pi}_s \| \pi_s) \right].$$

Thus, the proof is complete. □

Now, we start to prove Theorem 4.4. Here, for simplicity, we keep using the function ψ_s and ψ defined in Eq. (32) and (33). However, we ignore the superscript Φ since in this concrete example, we only take Φ to be the negative entropy.

Theorem 4.4 (Sublinear Convergence of SAC)**.** *Consider running Algorithm 1 for entropy-regularized reinforcement learning with initial policy $\pi^{(0)}$, regularization strength τ, initial distribution $\rho \in \Delta(\mathcal{S})$ and Φ being the negative entropy restricted on $\Delta(\mathcal{A})$. Suppose Assumptions (A1), (A2) and (A3) hold and the step sizes satisfy $\eta_k = \eta \leq \frac{1}{\tau \vartheta_\rho}$ for any k. Then, for any comparator policy $\pi^\star$, we have*

$$\frac{1}{K}\sum_{k=0}^{K-1}\left(V_{\tau,\rho}^{(k)} - V_{\tau,\rho}^\star\right) \;\leq\; \frac{1}{K}\left(\frac{D_0^\star}{(1-\gamma)\eta} + \frac{V_{\tau,d_\rho^\star}^{(0)}}{1-\gamma}\right) + \frac{\vartheta_\rho \psi(\epsilon_{\text{actor}}) + (2-\gamma)\vartheta_\rho \epsilon_{\text{critic}}}{(1-\gamma)^2},$$

where $D_0^\star = \mathbb{E}_{s\sim d_\rho^\star}\left[D_{\mathrm{KL}}\left(\pi_s^\star \,\|\, \pi_s^{(0)}\right)\right]$ and $\psi(x) = (1 + C_\rho)\left(x + \sqrt{2x}\right)$ for $x \geq 0$.

Proof. First, it is straightforward to check that Lemma D.5 still holds under entropy-regularized reinforcement learning. Then, fix some $s \in \mathcal{S}$ and $k < K$, similar to the proof of Theorem D.6, just like Eq. (34), we also have

$$\Delta_{\tau,s}^{(k)} \stackrel{\text{def}}{=} \eta\left\langle \widehat{Q}_{\tau,s}^{(k)}, \pi_s^{(k+1)} - \pi_s^{(k)}\right\rangle - \psi_s\left(D_{\mathrm{KL}}\left(\pi_s^{(k+1)} \,\middle\|\, \pi_s^{(k)}\exp\left(-\eta \widehat{Q}_{\tau,s}^{(k)}\right)/Z_s^{(k)}\right)\right) \leq 0. \tag{38}$$

Then, by using Lemma D.5 with $\pi_s = \pi_s^\star$, the comparator policy, we have

$$\begin{aligned}&\eta\left\langle \widehat{Q}_{\tau,s}^{(k)}, \pi_s^{(k+1)} - \pi_s^\star\right\rangle + D_{\mathrm{KL}}\left(\pi_s^\star \,\|\, \pi_s^{(k+1)}\right) + D_{\mathrm{KL}}\left(\pi_s^{(k+1)} \,\|\, \pi_s^{(k)}\right) - D_{\mathrm{KL}}\left(\pi_s^\star \,\|\, \pi_s^{(k)}\right)\\ \leq&\psi_s\left(D_{\mathrm{KL}}\left(\pi_s^{(k+1)} \,\middle\|\, \pi_s^{(k)}\exp\left(-\eta \widehat{Q}_{\tau,s}^{(k)}\right)/Z_s^{(k)}\right)\right).\end{aligned}$$

Notice here the key difference to the proof of Theorem D.6 is that we do **not** drop the term $D_{\mathrm{KL}}\left(\pi_s^{(k+1)} \,\middle\|\, \pi_s^{(k)}\right)$.

By some algebraic rearrangement and taking expectation with respect to distribution $d_\rho^\star$, we then get

$$\begin{aligned}&\mathbb{E}_{s\sim d_\rho^\star}\left[\eta\left\langle \widehat{Q}_{\tau,s}^{(k)}, \pi_s^{(k+1)} - \pi_s^{(k)}\right\rangle - \psi_s\left(D_{\mathrm{KL}}\left(\pi_s^{(k+1)} \,\middle\|\, \pi_s^{(k)}\exp\left(-\eta \widehat{Q}_{\tau,s}^{(k)}\right)/Z_s^{(k)}\right)\right)\right]\\ &\quad + \eta\mathbb{E}_{s\sim d_\rho^\star}\left[\left\langle \widehat{Q}_{\tau,s}^{(k)}, \pi_s^{(k)} - \pi_s^\star\right\rangle\right] + \mathbb{E}_{s\sim d_\rho^\star}\left[D_{\mathrm{KL}}\left(\pi_s^{(k+1)} \,\|\, \pi_s^{(k)}\right)\right] \leq D_k^\star - D_{k+1}^\star,\end{aligned} \tag{39}$$

where $D_k^\star = \mathbb{E}_{s\sim d_\rho^\star}\left[D_{\mathrm{KL}}\left(\pi_s^\star \,\middle\|\, \pi_s^{(k)}\right)\right]$.

For the first expectation above, we have

$$\begin{aligned}&\mathbb{E}_{s\sim d_\rho^\star}\left[\Delta_{\tau,s}^{(k)}\right]\\ \stackrel{\text{(i)}}{\geq}&\frac{1}{1-\gamma}\mathbb{E}_{s\sim d_{d_\rho^\star}^{(k+1)}}\left[\Delta_{\tau,s}^{(k)}\right]\\ =&\frac{\eta}{1-\gamma}\mathbb{E}_{s\sim d_{d_\rho^\star}^{(k+1)}}\left[\left\langle Q_{\tau,s}^{(k)}, \pi_s^{(k+1)} - \pi_s^{(k)}\right\rangle\right] + \frac{\eta}{1-\gamma}\mathbb{E}_{s\sim d_{d_\rho^\star}^{(k+1)}}\left[\left\langle \widehat{Q}_{\tau,s}^{(k)} - Q_{\tau,s}^{(k)}, \pi_s^{(k+1)} - \pi_s^{(k)}\right\rangle\right]\\ &\quad - \frac{\vartheta_\rho}{1-\gamma}\mathbb{E}_{s\sim\nu^{(k)}}\left[\psi_s\left(D_{\mathrm{KL}}\left(\pi_s^{(k+1)} \,\middle\|\, \pi_s^{(k)}\exp\left(-\eta \widehat{Q}_{\tau,s}^{(k)}\right)/Z_s^{(k)}\right)\right)\right]\\ &\hfill\text{(By Assumption (A2))}\\ \geq&\frac{\eta}{1-\gamma}\mathbb{E}_{s\sim d_{d_\rho^\star}^{(k+1)}}\left[\left\langle Q_{\tau,s}^{(k)}, \pi_s^{(k+1)} - \pi_s^{(k)}\right\rangle\right] - \frac{\eta\epsilon_{\text{critic}}}{1-\gamma} - \frac{\vartheta_\rho}{1-\gamma}\psi(\eta\epsilon_{\text{actor}})\\ &\hfill\text{(By Assumption (A1), (A3) and concavity of } \psi_s\text{.)}\\ =&\eta\left(V_{\tau,d_\rho^\star}^{(k+1)} - V_{\tau,d_\rho^\star}^{(k)}\right) - \eta\tau\mathbb{E}_{s\sim d_{d_\rho^\star}^{(k+1)}}\left[D_{\mathrm{KL}}\left(\pi_s^{(k+1)} \,\|\, \pi_s^{(k)}\right)\right] - \frac{\eta\vartheta_\rho\epsilon_{\text{critic}}}{1-\gamma} - \frac{\vartheta_\rho\psi(\eta\epsilon_{\text{actor}})}{1-\gamma}\\ &\hfill\text{(By Lemma 4.3, the modified performance difference lemma.)}\\ \geq&\eta\left(V_{\tau,d_\rho^\star}^{(k+1)} - V_{\tau,d_\rho^\star}^{(k)}\right) - \eta\tau\vartheta_\rho\mathbb{E}_{s\sim d_\rho^\star}\left[D_{\mathrm{KL}}\left(\pi_s^{(k+1)} \,\|\, \pi_s^{(k)}\right)\right] - \frac{\eta\vartheta_\rho\epsilon_{\text{critic}}}{1-\gamma} - \frac{\vartheta_\rho\psi(\eta\epsilon_{\text{actor}})}{1-\gamma}\end{aligned}$$

Here, the above inequality (i) holds because Eq. (38) holds and we have $d^{(k+1)}_{d^\star_\rho,s} \geq (1-\gamma)d^\star_{\rho,s}$ for any $s \in \mathcal{S}$ as introduced in Section 2.

Then, for the second expectation in Eq. (39), we can similarly apply Lemma 4.3 and obtain

$$\begin{aligned}\eta\mathbb{E}_{s\sim d^\star_\rho}\left[\left\langle \widehat{Q}^{(k)}_{\tau,s}, \pi^{(k)}_s - \pi^\star_s\right\rangle\right] =& \eta\mathbb{E}_{s\sim d^\star_\rho}\left[\left\langle Q^{(k)}_{\tau,s}, \pi^{(k)}_s - \pi^\star_s\right\rangle\right] + \eta\mathbb{E}_{s\sim d^\star_\rho}\left[\left\langle \widehat{Q}^{(k)}_{\tau,s} - Q^{(k)}_{\tau,s}, \pi^{(k)}_s - \pi^\star_s\right\rangle\right] \\ \geq& (1-\gamma)\eta\left(V^{(k)}_{\tau,\rho} - V^\star_{\tau,\rho}\right) + (1-\gamma)\eta\tau D^\star_k - \eta\vartheta_\rho\epsilon_{\text{critic}}.\end{aligned}$$

By plugging these bounds back into Eq. (39), we then have

$$\begin{aligned}(1-\gamma)\left(V^{(k)}_{\tau,\rho} - V^\star_{\tau,\rho}\right) \leq& \frac{D^\star_k}{\eta} - \frac{D^\star_{k+1}}{\eta} + V^{(k)}_{\tau,d^\star_\rho} - V^{(k+1)}_{\tau,d^\star_\rho} + \frac{(2-\gamma)\vartheta_\rho\epsilon_{\text{critic}}}{1-\gamma} + \frac{\vartheta_\rho\psi(\eta\epsilon_{\text{actor}})}{(1-\gamma)\eta} \\ &+ \left(\tau\vartheta_\rho - \frac{1}{\eta}\right)\mathbb{E}_{s\sim d^\star_\rho}\left[D_{\text{KL}}\left(\pi^{(k+1)}_s \,\Big\|\, \pi^{(k)}_s\right)\right] - (1-\gamma)\tau D^\star_k \\ \leq& \frac{D^\star_k}{\eta} - \frac{D^\star_{k+1}}{\eta} + V^{(k)}_{\tau,d^\star_\rho} - V^{(k+1)}_{\tau,d^\star_\rho} + \frac{(2-\gamma)\vartheta_\rho\epsilon_{\text{critic}}}{1-\gamma} + \frac{\vartheta_\rho\psi(\eta\epsilon_{\text{actor}})}{(1-\gamma)\eta} \\ &\text{(By taking } \eta \leq \tfrac{1}{\tau\vartheta_\rho} \text{ and noticing KL divergence is non-negative.)}\end{aligned}$$

Finally, by noticing that $\frac{\psi(\eta\epsilon_{\text{actor}})}{\eta} \leq \psi(\epsilon_{\text{actor}})$ and by taking sum from $k=0$ to $K-1$, we can get

$$\frac{1}{K}\sum_{k=0}^{K-1}\left(V^{(k)}_{\tau,\rho} - V^\star_{\tau,\rho}\right) \leq \frac{1}{K}\left(\frac{D^\star_0}{(1-\gamma)\eta} + \frac{V^{(0)}_{\tau,d^\star_\rho}}{1-\gamma}\right) + \frac{\vartheta_\rho\psi(\epsilon_{\text{actor}}) + (2-\gamma)\vartheta_\rho\epsilon_{\text{critic}}}{(1-\gamma)^2}.$$

□

F Implementation Details

F.1 Algorithm Details

The implementations of DAPO-KL, AMPO and MAMPO are based on modifying the actor loss in SAC while keeping other parts unchanged. Therefore, we will first present the pseudocode of SAC and then give modified actor losses for DAPO-KL, AMPO and MAMPO.

F.1.1 SAC

The pseudocode of SAC is given in Algorithm 2.

Here, $J(\tau,\theta)$ and $J_q(\phi_i, \mathcal{B}, \phi_{\text{targ},1}, \phi_{\text{targ},2})$ in line 10 and 11 represent the loss functions to update regularization parameter τ and q-value networks, respectively. More details of these two loss functions can be found in Haarnoja et al. [2018b].

F.1.2 DAPO-KL

To implement DAPO-KL, we will basically replace the update rule in 14 of Algorithm 2 by DAPO-KL's update rule. To do this, we first need to rewrite DAPO-KL's update rule in Eq. (24) in terms of $q^\pi_\tau = Q^\pi_\tau(s,a) + \tau\log\pi(a \mid s)$. In particular, we have

$$\begin{aligned}\theta^{(k+1)} \in& \arg\min_\theta \mathbb{E}_{s\sim d^{(k)}_\rho}\left[D_{\text{KL}}\left(\pi^\theta_s \,\middle\|\, \frac{\pi^{(k)}_s \exp\left(\eta_k Q^{(k)}_{\tau,s}\right)}{Z^{(k)}_s}\right)\right] \\ =& \arg\min_\theta \mathbb{E}_{s\sim d^{(k)}_\rho}\left[D_{\text{KL}}\left(\pi^\theta_s \,\middle\|\, \frac{\pi^{(k)}_s \exp\left(\eta_k q^{(k)}_{\tau,s} - \eta_k\tau\log\pi^{(k)}_s\right)}{Z^{(k)}_s}\right)\right] \\ =& \arg\min_\theta \mathop{\mathbb{E}}_{\substack{s\sim d^{(k)}_\rho \\ a\sim\pi^\theta_s}}\left[\log\pi^\theta(a \mid s) - (1-\eta_k\tau)\log\pi^{(k)}(a \mid s) - \eta_k q^{(k)}_\tau(s,a)\right] \\ &\text{(By ignoring normalization constants.)}\end{aligned}$$

$$= \arg\min_{\theta} \mathop{\mathbb{E}}_{\substack{s \sim d_\rho^{(k)} \\ a \sim \pi_s^\theta}} \left[\tau \log \pi^\theta(a \mid s) - (1-\beta)\tau \log \pi^{(k)}(a \mid s) - \beta q_\tau^{(k)}(s,a)\right],$$

where $\beta \stackrel{\text{def}}{=} \eta_k \tau < 1$. We can see that the update rule exactly becomes the SAC's update rule when $\beta = 1$, which means to have $\eta_k = \frac{1}{\tau}$, consistent with our derivation in Section 3.2.

Algorithm 2 Soft Actor-Critic (SAC) [Haarnoja et al., 2018b]

1: **Input:** Initial policy network parameter θ; initial q-value network parameters ϕ_1, ϕ_2; replay buffer $\mathcal{D}$; learning rates λ_q, λ_π, λ; target mixture weight $\omega \in (0,1)$; initial regularization power $\tau > 0$; number of gradient steps per iteration m
2: Set $\phi_{\text{targ},1}^{(0)} \leftarrow \phi_1$ and $\phi_{\text{targ},2}^{(0)} \leftarrow \phi_2$
3: Initialize $k \leftarrow 0$
4: **while** *not done* **do**
5: Observe state s and take action $a \sim \pi_\theta(\cdot \mid s)$
6: Observe next state s', reward r, done signal d and add (s,a,s',r,d) to buffer $\mathcal{D}$
// $\{d = 1$ if s' is a terminal state; otherwise, $d = 0\}$
7: If s' is a terminal state, reset the environment
8: **if** *it's time to update* **then**
9: Randomly sample a batch of transitions $\mathcal{B} = \{(s,a,r,s',d)\} \subseteq \mathcal{D}$
10: Update regularization parameter by

$$\tau^{(k+1)} \leftarrow \tau^{(k)} - \lambda \nabla_\tau J(\tau, \theta^{(k)})$$

11: Update q-value networks by

$$\phi_i^{(k+1)} \leftarrow \phi_i^{(k)} - \lambda_q \nabla_{\phi_i} J_q(\phi_i, \mathcal{B}, \phi_{\text{targ},1}^{(k)}, \phi_{\text{targ},2}^{(k)}, \theta^{(k)}, \tau^{(k+1)}), \quad i = 1,2$$

12: Set $\theta_0^{(k+1)} \leftarrow \theta^{(k)}$
13: **for** $j = 1, \ldots, m$ **do**
14: Update policy network by

$$\theta_j^{(k+1)} \leftarrow \theta_{j-1}^{(k+1)} - \lambda_\pi \nabla_\theta \mathop{\mathbb{E}}_{\substack{s \sim \mathcal{B} \\ a \sim \pi^\theta(\cdot \mid s)}} \left[\tau^{(k+1)} \log \pi^\theta(a \mid s) - \min_{i=1,2} q^{\phi_i^{(k+1)}}(s,a)\right]\Bigg|_{\theta = \theta_{j-1}^{(k+1)}}$$

15: **end for**
16: Set $\theta^{(k+1)} \leftarrow \theta_m^{(k+1)}$
17: Update target networks with

$$\phi_{\text{targ},i}^{(k+1)} \leftarrow \omega \phi_{\text{targ},i}^{(k)} + (1-\omega)\phi_i^{(k+1)}, \quad i = 1,2$$

18: Update $k \leftarrow k+1$
19: **end if**
20: **end while**

Therefore, to implement DAPO-KL, we replace the update rule in line 14 of Algorithm 2 by

$$\theta_j^{(k+1)} \leftarrow \theta_{j-1}^{(k+1)} - \lambda_\pi \nabla_\theta \mathop{\mathbb{E}}_{\substack{s \sim \mathcal{B} \\ a \sim \pi_s^\theta}} \Bigg[\tau^{(k+1)} \log \pi^\theta(a \mid s) - (1-\beta)\tau^{(k+1)} \log \pi^{\theta^{(k)}}(a \mid s) - \beta \min_{i=1,2} q^{\phi_i^{(k+1)}}(s,a)\Bigg]\Bigg|_{\theta = \theta_{j-1}^{(k+1)}},$$

where β is a user-specified hyperparameter. Note that we use the standard reparameterization trick to compute the above gradient [Kingma and Welling, 2013].

F.1.3 AMPO

To implement AMPO in Alfano et al. [2023], we will need to replace the update rule in line 14 of Algorithm 2 by AMPO's loss in Eq. (16) in a more concrete form. That is, we have

$$\begin{aligned}\theta^{(k+1)} &\in \arg\min_{\theta} \mathbb{E}_{s\sim d_\rho^{(k)}} \left[\left\| f_s^\theta - \left(\log \pi_s^{(k)} + \eta_k Q_s^{(k)}\right) \right\|_2^2 \right] \\ &= \arg\min_{\theta} \mathbb{E}_{s\sim d_\rho^{(k)}} \left[\left\| f_s^\theta - (1-\eta_k\tau)\log \pi_s^{(k)} - \eta_k q_{\tau,s}^{(k)} \right\|_2^2 \right].\end{aligned} \tag{40}$$

Here, f^θ should be the exponent of a Gaussian distribution. Therefore, if $g^\theta : \mathcal{S} \mapsto \mathbb{R}^{2n_\mathcal{A}}$ is the policy network, where $n_\mathcal{A}$ is the dimension of the action space. Then, we have

$$f^\theta(s,a) = -\sum_{i=1}^{n_\mathcal{A}} \frac{\left(g^\theta(s)_i\right)^2 - 2a_i g^\theta(s)_i}{2\left(g^\theta(s)_{n_\mathcal{A}+i}\right)^2}. \tag{41}$$

Therefore, to implement AMPO, we replace the update rule in line 14 of Algorithm 2 by

$$\theta_j^{(k+1)} \leftarrow \theta_{j-1}^{(k+1)} - \lambda_\pi \nabla_\theta \mathop{\mathbb{E}}_{\substack{s\sim\mathcal{B}\\ a\sim \mathsf{Unif}(\mathcal{A})}} \left[\left(f^\theta(s,a) - (1-\eta\tau^{(k+1)})\log \pi^{\theta^{(k)}}(a \mid s) - \eta \min_{i=1,2} q^{\phi_i^{(k)}}(s,a) \right)^2 \right] \Bigg|_{\theta=\theta_{j-1}^{(k+1)}},$$

where η is the mirror descent learning rate.

F.1.4 MAMPO

As discussed in Section 5, MAMPO tries to optimize

$$\theta^{(k+1)} \in \arg\min_{\theta} \mathbb{E}_{s\sim d_\rho^{(k)}} \left[\left\| f_s^\theta - \left(\pi_s^{(k)} + \eta_k Q_s^{(k)}\right) \right\|_2^2 \right],$$

where $f^\theta(s,a)$ is defined in Eq. (41). Therefore, to implement MAMPO, we replace the update rule in line 14 of Algorithm 2 by

$$\theta_j^{(k+1)} \leftarrow \theta_{j-1}^{(k+1)} - \lambda_\pi \nabla_\theta \mathop{\mathbb{E}}_{\substack{s\sim\mathcal{B}\\ a\sim \mathsf{Unif}(\mathcal{A})}} \left[\left(f^\theta(s,a) - \pi^{\theta^{(k)}}(a \mid s) - \eta \min_{i=1,2} q^{\phi_i^{(k)}}(s,a) + \eta\tau^{(k+1)} \log \pi^{\theta^{(k)}}(a \mid s) \right)^2 \right] \Bigg|_{\theta=\theta_{j-1}^{(k+1)}}.$$

F.2 Hyperparameter Settings

We use the implementation of SAC from the *Stable Baseline 3* under the MIT license [Raffin et al., 2021]. Then, we implement DAPO-KL, AMPO-KL, and MAMPO as modifications of its SAC's implementation. All model trainings were completed on 8 NVIDIA V100 GPUs in cluster.

We use SAC's default hyperparameters on all environments for both SAC and DAPO-KL, while AMPO-KL and MAMPO contain some tuning. Full hyperparameter details are provided in Table 1.

Particularly for hyperparameter β in DAPO-KL, we take different values for different tasks as shown in Table 2.

G Additional Experiment Results on AMPO

In this section, we first introduce the original version of AMPO proposed in Alfano et al. [2023], which is slightly different from what we have in Eq. (40). Then, we present a partial record of our efforts in tuning AMPO, which shows the difficulty of using this algorithm in practical scenario. Nevertheless, we retain the possibility that our implementation of AMPO may not be the optimal.

Table 1: Hyperparameters of all algorithms

Hyperparameter	SAC	**DAPO-KL**	AMPO	MAMPO
Adam learning rate	3×10^{-4}	3×10^{-4}	2×10^{-5}	3×10^{-4}
MD learning rate (η)	NA	NA	1.0	1.0
Entropy regularization (τ)	auto*	auto*	0	0
Number of hidden layers	2			
Hidden layer size	256			
Batch size	256			
Discount factor (γ)	0.99			
Target mixture weight (ω)	0.005			
Replay buffer size	1×10^{6}			

* Being "auto" in entropy regularization means to use the update rule at line 10 of Algorithm 2 to automatically adjust τ.

Table 2: Values of hyperparameter β in DAPO-KL for different MuJoCo tasks.

Environments	HalfCheetah-v4	Hopper-v4	Walker2d-v4	Ant-v4
β	0.7	0.6	0.4	0.7

G.1 Variants of AMPO

The original version of AMPO proposed in Alfano et al. [2023] is given as

$$\theta^{(k+1)} \in \arg\min_{\theta} \mathbb{E}_{s \sim d_\rho^{(k)}} \left[\left\| f_s^\theta - \left(f_s^{(k)} + \eta_k Q_s^{(k)} \right) \right\|_2^2 \right]. \tag{42}$$

While seemingly different from Eq. (40), these two are essentially the same from a theoretical perspective. To see this, as discussed in Example 2.3, when Φ is the negative entropy restricted on $\Delta(\mathcal{A})$, we can freely take $\nabla\Phi(\pi)$ to be any vector in $\partial\Phi(\pi)$ while the corresponding Bregman divergence D_Φ is still well-defined. In particular, we have $\partial\Phi(\pi) = \{\log \pi + c\mathbf{1} \mid c \in \mathbb{R}\}$ with $\mathbf{1} = [1 \quad \cdots \quad 1]^\top$. As a result, since the difference between f_s^θ and $\log \pi_s^{(k)}$ is only an action-independent normalization constant, Eq. (40) and Eq. (42) are theoretically equivalent.[3]

Nevertheless, Eq. (40) and Eq. (42) may still be empirically different since the constant difference can still affect the L_2-loss minimization. Therefore, we consider and empirically compare these two different theoretically equivalent variants of AMPO-KL.[4]

G.2 Comparison between MAMPO and AMPO-KL

Here, we provide a comparison between MAMPO and the two variants of AMPO-KL in Fig. 2, where both variants use the same set of hyperparameters as given in Table 1.

We can see that both variants of AMPO-KL almost cannot learn anything non-trivial in all tasks.

G.3 AMPO-KL under Different Hyperparameters

Finally, we also provide a performance comparison of variants of AMPO-KL under different hyperparameter settings, where we only show the final-step performance under each setting, given in Table 3, 4, 5 and 6. Nevertheless, we can easily see that AMPO-KL still cannot learn anything non-trivial under all of these settings.

[3] While Alfano et al. [2023] claims to obtain Eq. (42) by taking Φ to be the negative entropy on $\mathbb{R}_+^{|\mathcal{A}|}$, this is not an appropriate argument because such a choice of Φ will enforce $\nabla\Phi(\pi) = \log \pi + \mathbf{1}$, excluding the freedom of choosing action-independent constant.

[4] We use the variant in Eq. (40) in all previous experiments.

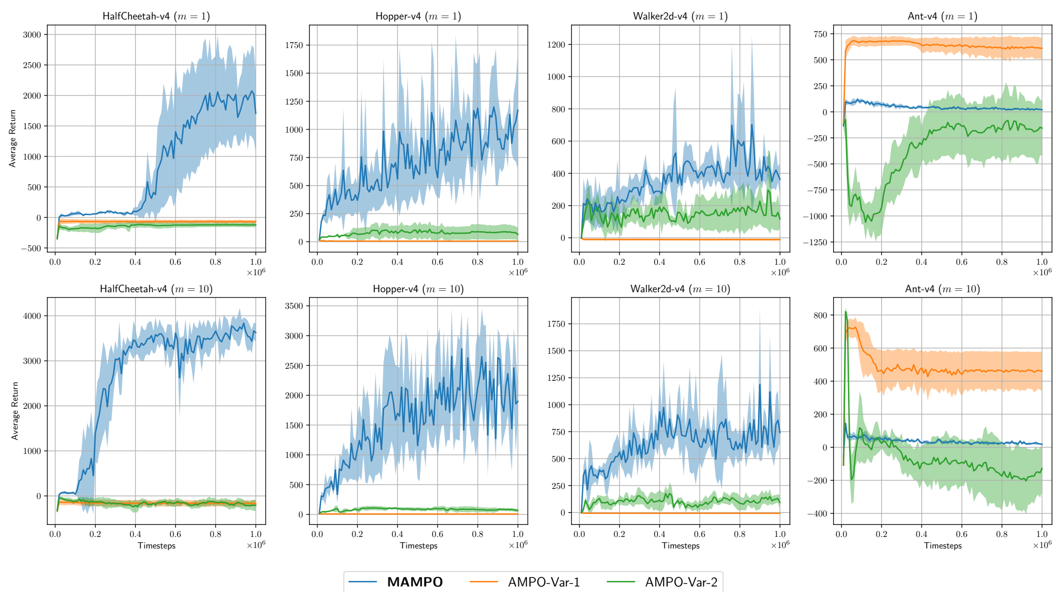

Figure 2: Comparison under $m = 1$ and $m = 10$ gradient steps per iteration between MAMPO and variants of AMPO-KL. Here, "AMPO-Var-1" refers to Eq. (40) and "AMPO-Var-2" refers to Eq. (42). Each curve is averaged over 5 different random seeds and the shaded area represents the 95% confidence interval.

Table 3: Final-step performance of AMPO-Var-1 (Eq. (40)) in HalfCheetah-v4 with entropy regularization ($\tau = 1.0$). Each data point is averaged over 3 different random seeds and $\pm$ represents the 95% confidence interval.

	$\text{lr} = 5 \times 10^{-6}$	$\text{lr} = 1 \times 10^{-5}$	$\text{lr} = 5 \times 10^{-5}$	$\text{lr} = 1 \times 10^{-4}$
$\eta_k = 0.1$	-94.08 ± 37.82	-77.18 ± 16.73	-3.33 ± 0.17	-178.56 ± 41.57
$\eta_k = 1$	-79.8 ± 47.96	-61.83 ± 20.85	-4.19 ± 0.49	-14.93 ± 11.05
$\eta_k = 10$	-27.83 ± 28.96	-201.02 ± 71.98	-8.37 ± 0.33	-7.58 ± 0.73
$\eta_k = 100$	-220.2 ± 124.88	-210.46 ± 168.3	-8.12 ± 0.35	-7.36 ± 0.75

Table 4: Final-step performance of AMPO-Var-1 (Eq. (40)) in HalfCheetah-v4 without entropy regularization ($\tau = 0$). Each data point is averaged over 3 different random seeds and $\pm$ represents the 95% confidence interval.

	$\text{lr} = 5 \times 10^{-6}$	$\text{lr} = 1 \times 10^{-5}$	$\text{lr} = 5 \times 10^{-5}$	$\text{lr} = 1 \times 10^{-4}$
$\eta_k = 0.1$	-131.27 ± 62.23	-129.24 ± 51.24	-123.17 ± 63.02	-109.34 ± 84.59
$\eta_k = 1$	-93.94 ± 46.46	-95.82 ± 49.19	-83.12 ± 59.42	-97.82 ± 47.88
$\eta_k = 10$	-50.57 ± 45.8	-98.75 ± 24.9	-81.51 ± 29.18	-57.17 ± 39.36
$\eta_k = 100$	-301.4 ± 128.66	-295.53 ± 63.03	-196.38 ± 136.29	-255.11 ± 143.2

Table 5: Final-step performance of AMPO-Var-2 (Eq. (42)) in HalfCheetah-v4 with entropy regularization ($\tau = 1.0$). Each data point is averaged over 3 different random seeds and $\pm$ represents the 95% confidence interval.

	$\text{lr} = 5 \times 10^{-6}$	$\text{lr} = 1 \times 10^{-5}$	$\text{lr} = 5 \times 10^{-5}$	$\text{lr} = 1 \times 10^{-4}$
$\eta_k = 0.1$	-3.56 ± 0.29	-3.48 ± 0.52	-3.39 ± 0.22	-3.59 ± 0.22
$\eta_k = 1$	-4.34 ± 0.47	-4.24 ± 0.32	-4.29 ± 0.26	-4.22 ± 0.34
$\eta_k = 10$	-8.38 ± 0.23	-8.33 ± 0.12	-8.4 ± 0.29	-8.08 ± 0.57
$\eta_k = 100$	41.59 ± 79.77	-8.38 ± 0.1	-8.33 ± 0.56	-8.36 ± 0.52

Table 6: Final-step performance of AMPO-Var-2 (Eq. (42)) in HalfCheetah-v4 without entropy regularization ($\tau = 0$). Each data point is averaged over 3 different random seeds and $\pm$ represents the 95% confidence interval.

	$\text{lr} = 5 \times 10^{-6}$	$\text{lr} = 1 \times 10^{-5}$	$\text{lr} = 5 \times 10^{-5}$	$\text{lr} = 1 \times 10^{-4}$
$\eta_k = 0.1$	-120.73 ± 34.97	-178.62 ± 34.49	-174.38 ± 67.63	-144.4 ± 54.12
$\eta_k = 1$	-87.45 ± 5.28	-121.66 ± 56.37	-129.73 ± 43.94	-157.94 ± 49.55
$\eta_k = 10$	-534.72 ± 205.25	-237.98 ± 127.79	-176.42 ± 235.9	-199.77 ± 12.45
$\eta_k = 100$	-341.7 ± 94.17	139.05 ± 118.93	-271.07 ± 149.01	-411.67 ± 56.92