A Novel Framework for Policy Mirror Descent with General parameterization and Linear Convergence

Carlo Alfano
Department of Statistics
University of Oxford
carlo.alfano@stats.ox.ac.uk

Rui Yuan
LTCI, Télécom Paris
Institut Polytechnique de Paris
yy42606r@gmail.com

Patrick Rebeschini
Department of Statistics
University of Oxford

Abstract

Modern policy optimization methods in reinforcement learning, such as Trust Region Policy Optimization and Proximal Policy Optimization, owe their success to the use of parameterized policies. However, while theoretical guarantees have been established for this class of algorithms, especially in the tabular setting, the use of general parameterization schemes remains mostly unjustified. In this work, we introduce a novel framework for policy optimization based on mirror descent that naturally accommodates general parameterizations. The policy class induced by our scheme recovers known classes, e.g., softmax, and generates new ones depending on the choice of mirror map. Using our framework, we obtain the first result that guarantees linear convergence for a policy-gradient-based method involving general parameterization. To demonstrate the ability of our framework to accommodate general parameterization schemes, we provide its sample complexity when using shallow neural networks and show that it represents an improvement upon the previous best results.

1 Introduction

Policy optimization is one of the most widely-used classes of algorithms for reinforcement learning (RL). Among policy optimization techniques, policy gradient (PG) methods [e.g., Williams and Peng, 1991, Sutton et al., 1999, Konda and Tsitsiklis, 2000, Baxter and Bartlett, 2001] are gradient-based algorithms that optimize the policy over a parameterized policy class and have emerged as a popular class of algorithms for RL [e.g., Kakade, 2002, Peters and Schaal, 2008, Bhatnagar et al., 2009, Mnih et al., 2016, Schulman et al., 2015a, 2017, Lan, 2022].

The design of gradient-based policy updates has been key to achieving empirical success in many settings, such as games [Berner et al., 2019] and autonomous driving [Shalev-Shwartz et al., 2016]. In particular, a class of PG algorithms that has proven successful in practice consists of building updates that include a hard constraint (e.g., a trust region constraint) or a penalty term ensuring that the updated policy does not move too far from the previous one. Two examples of algorithms belonging to this category are trust region policy optimization (TRPO) [Schulman et al., 2015a], which imposes a Kullback-Leibler (KL) divergence [Kullback

16th European Workshop on Reinforcement Learning (EWRL 2023).

and Leibler, 1951] constraint on its updates, and policy mirror descent (PMD) [e.g. Tomar et al., 2022, Lan, 2022, Xiao, 2022, Kuba et al., 2022, Vaswani et al., 2022], which applies mirror descent (MD) [Nemirovski and Yudin, 1983] to RL. Shani et al. [2020] propose a variant of TRPO that is actually a special case of PMD, thus linking TRPO and PMD.

From a theoretical perspective, motivated by the empirical success of PMD, there is now a concerted effort to develop convergence theories for PMD methods. For instance, it has been established that PMD converges linearly to the global optimum in the tabular setting by using a geometrically increasing step-size [Lan, 2022, Xiao, 2022], by adding entropy regularization [Cen et al., 2021], and more generally by adding convex regularization [Zhan et al., 2021]. Linear convergence of PMD has also been established for the negative entropy mirror map in the linear function approximation regime, i.e., for log-linear policies, either by adding entropy regularization [Cayci et al., 2021], or by using a geometrically increasing step-size [Chen and Theja Maguluri, 2022, Alfano and Rebeschini, 2022, Yuan et al., 2023]. The proofs of these results are based on specific policy parameterizations, i.e., tabular and log-linear, while PMD remains mostly unjustified for general policy parameterizations and mirror maps, leaving out important practical cases such as neural networks. In particular, it remains to be seen whether the theoretical results obtained for tabular policy classes transfer to this more general setting.

In this work, we introduce Approximate Mirror Policy Optimization (AMPO), a novel framework designed to incorporate general parameterization into PMD in a theoretically sound manner. In summary, AMPO is a MD-based method that recovers PMD in different settings, such as tabular MDPs, is capable of generating new algorithms by varying the mirror map, and is amenable to theoretical analysis for any parameterization class. Since the MD update can be viewed as a two-step procedure, i.e., a gradient update step on the dual space and a mapping step onto the probability simplex, our starting point is to define the policy class based on this second MD step (Definition 3.1). This policy class recovers the softmax policy class as a special case (Example 3.2) and accommodates any parameterization class, such as tabular, linear, or neural network parameterizations. We then develop an update procedure for this policy class based on MD and PG.

We provide an analysis of AMPO and establish theoretical guarantees that hold for any parameterization class and any mirror map. More specifically, we show that our algorithm enjoys quasi-monotonic improvements (Proposition 4.2), sublinear convergence when the step-size is non-decreasing, and linear convergence when the step-size is geometrically increasing (Theorem 4.3). To the best of our knowledge, AMPO is the first gradient-based policy optimization algorithm with linear convergence that can accommodate any parameterization class. Furthermore, the convergence rates hold for any choice of mirror map. The generality of our convergence results allows us not only to unify several current best-known results with specific policy parameterizations, i.e., tabular and log-linear, but also to achieve new state-of-the-art convergence rates with neural policies. Tables 1 and 2 in Appendix A.2 provide an overview of our results. We also refer to Appendix A.2 for a thorough literature review.

The key point of our analysis is Lemma 4.1, which is an application of the three-point descent lemma by Chen and Teboulle [1993, Lemma 3.2], typically used in the tabular setting, to the general parameterization setting. The application of this existing lemma is only possible thanks to our formulations of the policy class and the policy update, which allows us to keep track of the errors incurred by the algorithm (Proposition 4.2). The convergence rates of AMPO are obtained by building on Lemma 4.1 and leveraging the PMD proof techniques of Xiao [2022].

In addition, we show that for a large class of mirror maps, i.e., the ω-potential mirror maps in Definition 3.4, AMPO can be implemented in $\widetilde{\mathcal{O}}(|\mathcal{A}|)$ computations. We give two examples of mirror maps belonging to this class, Examples 3.5 and 3.6, that illustrate the versatility of our framework. Lastly, we examine the important case of shallow neural network parameterization. In this setting, we provide the sample complexity of AMPO, i.e., $\widetilde{\mathcal{O}}(\varepsilon^{-4})$ (Corollary 4.4), and show how it improves upon previous results.

2 Preliminaries

Let $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma, \mu)$ be a discounted Markov Decision Process (MDP), where $\mathcal{S}$ is a possibly infinite state space, $\mathcal{A}$ is a finite action space, $P(s'|s,a)$ is the transition probability from state s to s' under action a, $r(s,a) \in [0,1]$ is a reward function, γ is a discount factor, and μ is a target state distribution. The behavior of an agent on an MDP is then modeled by a *policy* $\pi \in (\Delta(\mathcal{A}))^{\mathcal{S}}$, where $a \sim \pi(\cdot \mid s)$ is the density of the distribution over actions at state $s \in \mathcal{S}$, and $\Delta(\mathcal{A})$ is the probability simplex over $\mathcal{A}$.

Given a policy π, let $V^\pi : \mathcal{S} \to \mathbb{R}$ denote the associated *value function*. Letting s_t and a_t be the current state and action at time t, the value function V^π is defined as the expected discounted cumulative reward with the initial state $s_0 = s$, namely,

$$V^\pi(s) := \mathbb{E}_{a_t \sim \pi(\cdot|s_t), s_{t+1} \sim P(\cdot|s_t,a_t)} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \middle| \pi, s_0 = s \right].$$

Now letting $V^\pi(\mu) := \mathbb{E}_{s\sim\mu}[V^\pi(s)]$, our objective is for the agent to find an optimal policy

$$\pi^\star \in \operatorname{argmax}_{\pi \in (\Delta(\mathcal{A}))^{\mathcal{S}}} V^\pi(\mu). \tag{1}$$

Similarly to the value function, for each pair $(s,a) \in \mathcal{S} \times \mathcal{A}$, the state-action value function, or *Q-function*, associated to a policy π is defined as

$$Q^\pi(s,a) := \mathbb{E}_{a_t \sim \pi(\cdot|s_t), s_{t+1} \sim P(\cdot|s_t,a_t)} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid \pi, s_0 = s, a_0 = a \right].$$

We also define the discounted state visitation distribution by

$$d_\mu^\pi(s) := (1-\gamma)\mathbb{E}_{s_0 \sim \mu} \left[\sum_{t=0}^{\infty} \gamma^t P(s_t = s \mid \pi, s_0) \right], \tag{2}$$

where $P(s_t = s \mid \pi, s_0)$ represents the probability of the agent being in state s at time t when following policy π and starting from s_0. The probability $d_\mu^\pi(s)$ represents the time spent on state s when following policy π.

The gradient of the value function $V^\pi(\mu)$ with respect to the policy can be easily expressed by the policy gradient theorem [Sutton et al., 1999]:

$$\nabla_s V^\pi(\mu) := \frac{\partial V^\pi(\mu)}{\partial \pi(\cdot|s)} = \frac{1}{1-\gamma} d_\mu^\pi(s) Q^\pi(s, \cdot). \tag{3}$$

2.1 Mirror descent

The first tools we recall from the MD framework are mirror maps and Bregman divergences [Bubeck, 2015, Chapter 4]. Let $\mathcal{Y} \subseteq \mathbb{R}^{|\mathcal{A}|}$ be a convex set. A *mirror map* $h : \mathcal{Y} \to \mathbb{R}$ is a strictly convex, continuously differentiable and essentially smooth function[1] such that $\nabla h(\mathcal{Y}) = \mathbb{R}^{|\mathcal{A}|}$. The convex conjugate of h, denoted by h^*, is given by

$$h^*(x^*) := \sup_{x \in \mathcal{Y}} \langle x^*, x \rangle - h(x), \quad x^* \in \mathbb{R}^{|\mathcal{A}|}.$$

The gradient of the mirror map $\nabla h : \mathcal{Y} \to \mathbb{R}^{|\mathcal{A}|}$ allows to map objects from the primal space $\mathcal{Y}$ to its dual space $\mathbb{R}^{|\mathcal{A}|}$, $x \mapsto \nabla h(x)$, and viceversa for ∇h^*, i.e., $x^* \mapsto \nabla h^*(x^*)$. In particular, from $\nabla h(\mathcal{Y}) = \mathbb{R}^{|\mathcal{A}|}$, we have: for all $(x, x^*) \in \mathcal{Y} \times \mathbb{R}^{|\mathcal{A}|}$,

$$x = \nabla h^*(\nabla h(x)) \quad \text{and} \quad x^* = \nabla h(\nabla h^*(x^*)). \tag{4}$$

Furthermore, the mirror map h induces a *Bregman divergence* [Bregman, 1967] , defined as

$$\mathcal{D}_h(x,y) := h(x) - h(y) - \langle \nabla h(y), x - y \rangle,$$

where $\mathcal{D}_h(x,y) \geq 0$ for all $x, y \in \mathcal{Y}$. We can now present the standard MD algorithm [Nemirovski and Yudin, 1983, Bubeck, 2015]. Let $\mathcal{X} \subseteq \mathcal{Y}$ be a convex set and $V : \mathcal{X} \to \mathbb{R}$ be

[1] h is essentially smooth if $\lim_{x \to \partial\mathcal{Y}} \|\nabla h(x)\|_2 = +\infty$, where $\partial\mathcal{Y}$ denotes the boundary of $\mathcal{Y}$.

a differentiable function. The MD algorithm can be formalized[2] as the following iterative procedure in order to solve the minimization problem $\min_{x\in\mathcal{X}} V(x)$: for all $t \geq 0$,

$$y^{t+1} = \nabla h(x^t) - \eta_t \nabla V(x)|_{x=x^t}, \tag{5}$$

$$x^{t+1} = \mathrm{Proj}^h_{\mathcal{X}}(\nabla h^*(y^{t+1})), \tag{6}$$

where η_t is set according to a step-size schedule $(\eta_t)_{t\geq 0}$ and $\mathrm{Proj}^h_{\mathcal{X}}(\cdot)$ is the *Bregman projection*

$$\mathrm{Proj}^h_{\mathcal{X}}(y) := \mathrm{argmin}_{x\in\mathcal{X}} \mathcal{D}_h(x, y). \tag{7}$$

Precisely, at time t, $x^t \in \mathcal{X}$ is mapped to the dual space through $\nabla h(\cdot)$, where a gradient step is performed as in (5) to obtain y^{t+1}. The next step is to map y^{t+1} back in the primal space using $\nabla h^*(\cdot)$. In case $\nabla h^*(y^{t+1})$ does not belong to $\mathcal{X}$, it is projected as in (6).

3 Approximate Mirror Policy Optimization

The starting point of our framework is the introduction of a novel parameterized policy class based on the Bregman projection expression recalled in (7).

Definition 3.1. Given a parameterized function class $\mathcal{F}^\Theta = \{f^\theta : \mathcal{S}\times\mathcal{A} \to \mathbb{R}, \theta \in \Theta\}$, a mirror map $h : \mathcal{Y} \to \mathbb{R}$, where $\mathcal{Y} \subseteq \mathbb{R}^{|\mathcal{A}|}$ is a convex set with $\Delta(\mathcal{A}) \subseteq \mathcal{Y}$, and $\eta > 0$, the *Bregman projected policy* class associated with $\mathcal{F}^\Theta$ and h consists of all the policies of the form:

$$\left\{\pi^\theta : \pi^\theta_s = \mathrm{Proj}^h_{\Delta(\mathcal{A})}(\nabla h^*(\eta f^\theta_s)),\ s \in \mathcal{S};\ \theta \in \Theta\right\},$$

where for all $s \in \mathcal{S}$, $\pi^\theta_s, f^\theta_s \in \mathbb{R}^{|\mathcal{A}|}$ denote vectors $[\pi^\theta(a|s)]_{a\in\mathcal{A}}$ and $[f^\theta(s,a)]_{a\in\mathcal{A}}$, respectively.

In this definition, the policy is induced by a mirror map h and a parameterized function f^θ, and is obtained by mapping f^θ to $\mathcal{Y}$ with the operator $\nabla h^*(\cdot)$, which may not be a well-defined probability distribution, and is thus projected on the convex probability simplex $\Delta(\mathcal{A})$. Note that the choice of h will be key to deriving convenient expressions for the policy π^θ. The Bregman projected policy class contains large families of policy classes. Below is an example of h that recovers widely used policy classes [Beck, 2017, Example 9.10].

Example 3.2 (Negative entropy mirror map)**.** If $\mathcal{Y} = \mathbb{R}^{|\mathcal{A}|}_+$ and h is the negative entropy mirror map, i.e., $h(\pi(\cdot|s)) = \sum_{a\in\mathcal{A}} \pi(a|s)\log(\pi(a|s))$, then $\mathrm{Proj}^h_{\Delta(\mathcal{A})}(\nabla h^*(\eta f^\theta_s))$ is equivalent to the following popular policy class

$$\left\{\pi^\theta : \pi^\theta_s = \frac{\exp(\eta f^\theta_s)}{\|\exp(\eta f^\theta_s)\|_1},\ s \in \mathcal{S};\ \theta \in \Theta\right\}, \tag{8}$$

where the exponential and the fraction are element-wise and $\|\cdot\|_1$ is ℓ_1 norm. In particular, when $f^\theta(s,a) = \theta_{s,a}$, the policy class (8) becomes tabular softmax policy; when f^θ is a linear function, (8) becomes log-linear policy; and when f^θ is a neural network, (8) becomes the neural policy class defined by Agarwal et al. [2021]. We refer to Appendix C.1 for details.

We now construct a policy mirror descent type algorithm to optimize V^{π^θ} over the Bregman projected policy class associated with a mirror map h and a parameterization class $\mathcal{F}^\Theta$ by adapting Section 2.1 to our setting. First, we use the following shorthand: at each time t, let $\pi^t := \pi^{\theta_t}$, $f^t := f^{\theta_t}$, $V^t := V^{\pi^t}$, $Q^t := Q^{\pi^t}$, and $d^t_\mu := d^{\pi^t}_\mu$. Further, for any function $y : \mathcal{S}\times\mathcal{A} \to \mathbb{R}$ and distribution v over $\mathcal{S}\times\mathcal{A}$, let $y_s := y(s,\cdot) \in \mathbb{R}^{|\mathcal{A}|}$ and $\|y\|^2_{L_2(v)} = \mathbb{E}_v[(y(s,a))^2]$. Ideally, we would like to execute the exact MD-based algorithm: for all $t \geq 0$ and for all $s \in \mathcal{S}$,

$$f^{t+1}_s = \nabla h(\pi^t_s) + \eta_t(1-\gamma)\nabla_s V^t(\mu)/d^t_\mu(s) \overset{(3)}{=} \nabla h(\pi^t_s) + \eta_t Q^t_s,\ ^3 \tag{9}$$

$$\pi^{t+1}_s = \mathrm{Proj}^h_{\Delta(\mathcal{A})}(\nabla h^*(\eta_t f^{t+1}_s)). \tag{10}$$

[2] See a different formulation of MD in (11) and in Appendix B (Lemma B.1).

Algorithm 1: Approximate Mirror Policy Optimization

Input: Initial policy π^0, mirror map h, parameterization class $\mathcal{F}^\Theta$, iteration number T, step-size schedule $(\eta_t)_{t\geq 0}$, state-action distribution sequence $(v_t)_{t\geq 0}$.

for $t = 0$ ***to*** $T-1$ **do**

1 Obtain $\theta^{t+1} \in \Theta$ such that

$$\theta^{t+1} \in \operatorname{argmin}_{\theta\in\Theta} \left\| f^\theta - Q^t - \eta_t^{-1}\nabla h(\pi^t)\right\|^2_{L_2(v_t)}.$$

2 Update

$$\pi_s^{t+1} = \operatorname*{argmin}_{\pi'\in\Delta(\mathcal{A})} \mathcal{D}_h(\pi', \nabla h^*(\eta_t f_s^{\theta^{t+1}})) = \operatorname{Proj}^h_{\Delta(\mathcal{A})}(\nabla h^*(\eta_t f_s^{t+1})),\ \forall s \in \mathcal{S}.$$

Here, (10) reflects our Bregman projected policy class 3.1. However, we usually cannot perform the update (9) exactly. In general, if f^θ belongs to a parameterized class $\mathcal{F}^\Theta$, there may not be any $\theta^{t+1} \in \Theta$ such that (9) is satisfied for all $s \in \mathcal{S}$.

To remedy this issue, we propose Approximate Mirror Policy Optimization (AMPO), described in Algorithm 1. At each iteration, AMPO consists of minimizing a surrogate loss function and projecting the result onto the simplex to obtain the updated policy. In particular, the surrogate loss in Line 1 of Algorithm 1 is a standard regression problem where we try to approximate $Q^t + \eta_t^{-1}\nabla h(\pi^t)$ with f^{t+1}, and has been studied extensively when f^θ is a neural network [Allen-Zhu et al., 2019a]. We can then readily use (10) to update π^{t+1} within the Bregman projected policy class defined in 3.1, which gives Line 2 of Algorithm 1.

Remark 3.3. Line 1 associates AMPO with the *compatible function approximation* framework developed by Sutton et al. [1999], Kakade [2002], Agarwal et al. [2021], as both frameworks define the updated parameters θ^{t+1} as the solution to a regression problem aimed at approximating the current Q-function Q^t. A crucial difference is that, Agarwal et al. [2021] approximate Q^t linearly w.r.t. $\nabla_\theta \log \pi^t$, while in Line 1 we approximate Q^t and the gradient of the mirror map of the previous policy with any function f^θ. In particular, the regression problem in Line 1 appears often in deep learning and has been well studied both theoretically [Allen-Zhu et al., 2019a] and empirically [Goodfellow et al., 2016], meaning that existing methods can be used to solve it. Furthermore, the regression problem proposed by Agarwal et al. [2021] depends on the distribution d_μ^t, while ours has no such constraint and allows off-policy updates involving an arbitrary distribution v^t. See Appendix A.3 for more details.

To better illustrate the novelty of our framework, we now give a comparison between AMPO and previous approximations of PMD [Vaswani et al., 2022, Tomar et al., 2022]. In both approaches, the algorithm provides an expression to be optimized. For AMPO this expression is the one in Line 1 of Algorithm 1, while, for instance, Vaswani et al. [2022] aim to maximize an expression equivalent to

$$\pi^{t+1} = \operatorname*{argmax}_{\pi^\theta\in\Pi(\Theta)} \mathbb{E}_{s\sim d_\mu^t}[\eta_t \langle Q_s^t, \pi_s^\theta\rangle - \mathcal{D}_h(\pi_s^\theta, \pi_s^t)], \tag{11}$$

where $\Pi(\Theta)$ is a given parameterized policy class. When the policy class $\Pi(\Theta)$ is the entire policy space $\Delta(\mathcal{A})^\mathcal{S}$, this is equivalent to the two-step procedure (9)-(10) thanks to the policy gradient theorem (3). A derivation of this observation is given in Appendix B for completeness. The improvement of AMPO over this type of update is twofold. Firstly, the parameterized policy class $\Pi(\Theta)$ is often non-convex with respect to θ in practice, which prevents the application of the three-point-descent lemma [Xiao, 2022], which relies on the convexity of the tabular parameterization. On the contrary, AMPO avoids this problem thanks to the definition of the Bregman projected policy class and the update in Line 2 of Algorithm 1, as we will see in the theoretical analysis. Secondly, AMPO involves a subroutine optimization procedure that is structurally different from the update in (11). Unlike the update in (11), our approach employs a standard regression procedure, which has been extensively researched and benefits from established solving methods. Additionally, we provide in Appendix A.4 a setting where the optimization problem in Line 1 of Algorithm 1 inherits curvature from the parameterization function, i.e., strong convexity, while the optimization problem in (11) does not.

[3]The update is (5) up to a scaling $(1-\gamma)/d_\mu^t(s)$ of η_t.

3.1 ω-potential mirror maps

In this section, we provide a class of mirror maps that allows to compute the Bregman projection in Line 2 with $\widetilde{\mathcal{O}}(|\mathcal{A}|)$ operations and simplifies the minimization problem in Line 1.

Definition 3.4 (ω-potential mirror map [Krichene et al., 2015])**.** For $a \in (-\infty, +\infty]$, $\omega \leq 0$, let an *ω-potential* be an increasing C^1-diffeomorphism $\phi : (-\infty, a) \to (\omega, +\infty)$ such that

$$\lim_{u \to -\infty} \phi(u) = \omega, \qquad \lim_{u \to a} \phi(u) = +\infty, \qquad \int_0^1 \phi^{-1}(u) du \leq \infty.$$

For any ω-potential ϕ, the associated mirror map h_ϕ, called *ω-potential mirror map*, is defined as

$$h_\phi(\pi_s) = \sum_{a \in \mathcal{A}} \int_1^{\pi(a|s)} \phi^{-1}(u) du.$$

Thanks to Krichene et al. [2015, Proposition 2], the policy π^{t+1} in Line 2 induced by the ω-potential mirror map can be obtained with $\widetilde{\mathcal{O}}(|\mathcal{A}|)$ computations and can be written as

$$\pi^{t+1}(a|s) = \sigma(\phi(\eta_t f^{t+1}(s,a) + \lambda_s^{t+1})) \quad \forall s \in \mathcal{S}, a \in \mathcal{A},$$

where $\lambda_s \in \mathbb{R}$ is a normalization factor to ensure $\sum_{a \in \mathcal{A}} \pi^{t+1}(a|s) = 1$ for all $s \in \mathcal{S}$, and $\sigma(z) = \max(z, 0)$ for $z \in \mathbb{R}$. We call this policy class the *ω-potential policy class*. The minimization problem in Line 1 is simplified to be

$$\theta^{t+1} \in \operatorname*{argmin}_{\theta \in \Theta} \left\| f^\theta - Q^t - \eta_t^{-1} \max(\eta_{t-1} f^t, \phi^{-1}(0) - \lambda_s^t) \right\|_{L_2(v_t)}^2. \tag{12}$$

We refer to Appendix C.2 for its derivation and an efficient implementation of the framework. This class of mirror maps allows AMPO to generate a wide range of algorithms by simply choosing an ω-potential ϕ. In fact, it recovers existing approaches to policy optimization, as we show in the next two examples.

Example 3.5 (Squared ℓ_2-norm)**.** If $\mathcal{Y} = \mathbb{R}^{|\mathcal{A}|}$ and ϕ is the identity function, then h_ϕ is the squared ℓ_2-norm, that is $h_\phi(\pi_s) = \|\pi_s\|^2 / 2$, Line 1 in Algorithm 1 becomes

$$\theta^{t+1} \in \operatorname*{argmin}_{\theta \in \Theta} \left\| f^\theta(s,a) - Q^t(s,a) - \eta_t^{-1} \pi^t(a|s) \right\|_{L_2(v_t)}^2, \tag{13}$$

and the policy update is given for all $s \in \mathcal{S}$ by

$$\pi_s^{t+1} = \mathrm{Proj}_{\Delta(\mathcal{A})}^{l_2}(\eta_t f_s^{t+1}), \tag{14}$$

where $\mathrm{Proj}_{\Delta(\mathcal{A})}^{l_2}$ represents the Euclidean projection on the probability simplex. In the tabular setting, where $\mathcal{S}$ and $\mathcal{A}$ are finite and $f^\theta(s,a) = \theta_{s,a}$, (13) can be solved exactly, and Equations (13) and (14) recover the projected Q-descent algorithm [Xiao, 2022]. For detailed derivations, we refer to Appendix C.2. As a by-product, we generalize the projected Q-descent algorithm from the tabular setting to a general parameterization class $\mathcal{F}^\Theta$, which is a novel algorithm in the RL literature.

Example 3.6 (Negative entropy)**.** If $\mathcal{Y} = \mathbb{R}_+^{|\mathcal{A}|}$ and $\phi(u) = \exp(u - 1)$, then h_ϕ is the negative entropy mirror map from Example 3.2 and Line 1 in Algorithm 1 becomes

$$\theta^{t+1} \in \operatorname*{argmin}_{\theta \in \Theta} \left\| f^{t+1} - Q^t - \frac{\eta_{t-1}}{\eta_t} f^t \right\|_{L_2(v_t)}^2. \tag{15}$$

Consequently, based on Example 3.2, we have $\pi_s^{t+1} \propto \exp(\eta_t f_s^{t+1})$ for all $s \in \mathcal{S}$. In this example, AMPO recovers tabular NPG [Shani et al., 2020] when $f^\theta(s,a) = \theta_{s,a}$, and NPG with log-linear polices [Yuan et al., 2023] when f^θ and Q^t are linear functions for all $t \geq 0$. We refer to Appendix C.2 for detailed derivations.

In addition to the ℓ_2-norm and the negative entropy, several other mirror maps that have been studied in the optimization literature fall into the class of ω-potential mirror maps, such as the Tsallis entropy [Orabona, 2020, Li and Lan, 2023] and the hyperbolic entropy [Ghai et al., 2020], as well as a generalization of the negative entropy proposed by Krichene et al. [2015] that has an exact solution for the associated Bregman projection. These examples illustrate how the ω-potential mirror map class recovers known methodologies and can be used to explore new algorithms in policy optimization. We leave the study of the application of these mirror maps in RL as future work, both from an empirical and theoretical point of view, and provide additional discussions and details in Appendix C.2.

4 Theoretical analysis

This section is devoted to the theoretical analysis of AMPO, which will be based on the following lemma. For convenience, denote $\mathcal{D}_{\bar{\pi}}^{\pi}(s) = \mathcal{D}_h(\pi_s, \bar{\pi}_s)$ for all $s \in \mathcal{S}$.

Lemma 4.1. *For any policies π and $\bar{\pi}$, for any function $f^\theta \in \mathcal{F}^\Theta$ and for $\eta > 0$, we have*

$$\langle \eta f_s^\theta - \nabla h(\bar{\pi}_s), \pi_s - \tilde{\pi}_s \rangle \leq \mathcal{D}_{\bar{\pi}}^{\pi}(s) - \mathcal{D}_{\bar{\pi}}^{\tilde{\pi}}(s) - \mathcal{D}_{\tilde{\pi}}^{\pi}(s), \quad \forall s \in \mathcal{S},$$

where $\tilde{\pi}$ is the Bregman projected policy induced by f^θ and h according to Definition 3.1, that is $\tilde{\pi}_s = \operatorname{argmin}_{\pi' \in \Delta(\mathcal{A})} \mathcal{D}_h(\pi', \nabla h^*(\eta f_s^\theta))$ for all $s \in \mathcal{S}$.

The proof of Lemma 4.1 is given in Appendix D.1. Lemma 4.1 describes a relation between any two policies and a policy belonging to the Bregman projected policy class associated with $\mathcal{F}^\Theta$ and h. It can be interpreted as an application of the three point descent lemma [Xiao, 2022] to the policy class we consider. Similar lemmas have been obtained and exploited for the negative entropy mirror map [Liu et al., 2019, Hu et al., 2022, Yuan et al., 2023].

Lemma 4.1 becomes useful when we set $\bar{\pi} = \pi^t$, $f^\theta = f^{t+1}$, $\eta = \eta_t$ and $\pi = \pi^t$ or $\pi = \pi^\star$. In particular, when $\eta_t f_s^{t+1} - \nabla h(\pi_s^t) \approx \eta_t Q_s^\pi$, Lemma 4.1 allows us to obtain telescopic sums and recursive relations, and to handle error terms efficiently, as we show in Appendix D. This is possible thanks to our two-step formulation (5)-(6) applied in Algorithm 1, while Lemma 4.1 cannot be applied to algorithms based on the update in (11) [Tomar et al., 2022, Vaswani et al., 2022] due to the non-convexity of the optimization problem.

4.1 Convergence for general policy parameterization

In this section, we consider the parameterization class $\mathcal{F}^\Theta$ and the fixed but arbitrary mirror map h. We show that AMPO enjoys quasi-monotonic improvement and sublinear or linear convergence, depending on the step-size schedule. The first step is to control the approximation error of AMPO by Assumption (A1) below.

(A1) (*Approximation error*). There exists $\varepsilon_{\text{approx}} \geq 0$ such that, for all times $t \geq 0$,

$$\mathbb{E}\big[\, \big\| f^{t+1} - Q^t - \eta_t^{-1} \nabla h(\pi^t) \big\|_{L_2(v_t)}^2 \,\big] \leq \varepsilon_{\text{approx}},$$

where $(v^t)_{t \geq 0}$ is a sequence of distributions over states and actions and the expectation is taken over the randomness of the algorithm that obtains f^{t+1}.

Assumption (A1) characterizes the loss incurred by Algorithm 1 in solving the regression problem in Line 1. When the step-size η_t is sufficiently large, Assumption (A1) measures how well f^{t+1} approximates the current Q-function Q^t. Hence, $\varepsilon_{\text{approx}}$ depends on both the accuracy of the policy evaluation method employed to obtain an estimate of Q^t [Sutton et al., 1998, Schulman et al., 2015b, Espeholt et al., 2018] and the error incurred by the function $f^\theta \in \mathcal{F}^\Theta$ that best approximates Q^t, that is the representation power of $\mathcal{F}^\Theta$. Later in Section 4.2, we show how to solve the minimization problem in Line 1 when $\mathcal{F}^\Theta$ is a class of shallow neural networks so that Assumption (A1) holds. We highlight that Assumption (A1) is weaker than the assumptions made by Agarwal et al. [2020, Chapter 6], since we do not constrain the minimization problem to be linear in the parameters (see (24)). We refer to Appendix E for a relaxed version of this assumption and further discussion.

As mentioned in Remark 3.3, the distribution v^t does not depend on the current policy π^t for all times $t \geq 0$. Therefore, Assumption (A1) allows off-policy policy updates and the use of replay buffers [Mnih et al., 2015]. To quantify how the choice of these distributions affects the error terms in the convergence rates, we introduce the following coefficient.

(A2) (*Concentrability coefficient*). There exists $C_v \geq 0$ such that, for all times t,

$$\mathbb{E}_{(s,a)\sim v^t}\left[\left(\frac{d_\mu^\pi(s)\pi(a|s)}{v^t(s,a)}\right)^2\right] \leq C_v,$$

whenever (d_μ^π, π) is either $(d_\mu^\star, \pi^\star)$, (d_μ^{t+1}, π^{t+1}), $(d_\mu^\star, \pi^t)$, or (d_μ^{t+1}, π^t).

The concentrability coefficient C_v quantifies how much the distribution v^t overlaps with the distributions $(d_\mu^\star, \pi^\star)$, (d_μ^{t+1}, π^{t+1}), $(d_\mu^\star, \pi^t)$ and (d_μ^{t+1}, π^t). Assumption (A2) highlights that the distribution v^t should have full support over the environment, in order to avoid large values of C_v. Assumption (A2) is weaker than the previous best known concentrability coefficient in Yuan et al. [2023, Assumption 9], in the sense that we have the full control over v^t. We refer to Appendix F for a more detailed discussion. We can now present our first result on the performance of Algorithm 1.

Proposition 4.2 (Quasi-monotonic updates)**.** *Let (A1), (A2) be true. We have, for all $t \geq 0$,*

$$\mathbb{E}\left[V^{t+1}(\mu) - V^t(\mu)\right] \geq -\frac{2\sqrt{C_v \varepsilon_{\text{approx}}}}{1-\gamma},$$

where the expectation is taken over the randomness of AMPO.

We refer to Appendix D.3 for a proof and a tighter bound. Proposition 4.2 ensures that an update of Algorithm 1 cannot lead to a performance degradation, up to an error term. The next assumptions concerns the coverage of the state space for the agent at each time t.

(A3) (*Distribution mismatch coefficient*). Let $d_\mu^\star := d_\mu^{\pi^\star}$. There exists $\nu_\mu \geq 0$ such that

$$\max_{s\in\mathcal{S}} \frac{d_\mu^\star(s)}{d_\mu^t(s)} \leq \nu_\mu, \quad \text{for all times } t \geq 0.$$

Since $d_\mu^t(s) \geq (1-\gamma)\mu(s)$ for all $s \in \mathcal{S}$, obtained from the definition of d_μ in (2), we have that

$$\max_{s\in\mathcal{S}} \frac{d_\mu^\star(s)}{d_\mu^t(s)} \leq \frac{1}{1-\gamma} \max_{s\in\mathcal{S}} \frac{d_\mu^\star(s)}{\mu(s)},$$

where assuming boundedness for the term on the right-hand side is standard in the literature on the PG convergence analysis [e.g., Zhang et al., 2020, Wang et al., 2020] and the NPG convergence analysis [e.g., Agarwal et al., 2021, Cayci et al., 2021, Xiao, 2022]. We refer to Appendix F for a more detailed discussion.

We also introduce the weighted Bregman divergence between the optimal policy $\pi^\star$ and the initial policy π^0 as $\mathcal{D}_0^\star = \mathbb{E}_{s\sim d_\mu^\star}[\mathcal{D}_h(\pi_s^\star, \pi_s^0)]$. We then have our main results below.

Theorem 4.3 (Convergence rates)**.** *Let (A1), (A2) and (A3) be true. If the step-size schedule is non-decreasing, i.e., $\eta_t \leq \eta_{t+1}$ for all $t \geq 0$, the iterates of Algorithm 1 satisfy: for every $T \geq 0$,*

$$V^\star(\mu) - \frac{1}{T}\sum_{t<T} \mathbb{E}\left[V^t(\mu)\right] \leq \frac{1}{T}\left(\frac{\mathcal{D}_0^\star}{(1-\gamma)\eta_0} + \frac{\nu_\mu}{1-\gamma}\right) + \frac{2(1+\nu_\mu)\sqrt{C_v\varepsilon_{\text{approx}}}}{1-\gamma}.$$

Furthermore, if the step-size schedule is geometrically increasing, i.e., satisfies

$$\eta_{t+1} \geq \frac{\nu_\mu}{\nu_\mu - 1}\eta_t \qquad \forall t \geq 0, \tag{16}$$

we have: for every $T \geq 0$,

$$V^\star(\mu) - \mathbb{E}\left[V^T(\mu)\right] \leq \frac{1}{1-\gamma}\left(1 - \frac{1}{\nu_\mu}\right)^T \left(1 + \frac{\mathcal{D}_0^\star}{\eta_0(\nu_\mu - 1)}\right) + \frac{2(1+\nu_\mu)\sqrt{C_v\varepsilon_{\text{approx}}}}{1-\gamma}.$$

Theorem 4.3 is, to the best of our knowledge, the first result that establishes linear convergence for a PG-based method involving general policy parameterization. For the same setting, it is also the first result that establishes $O(1/T)$ convergence without regularization. Lastly, it is the first result that provides a convergence rate for a PMD-based algorithm that allows any mirror map and non-tabular policies. We give here a brief discussion of Theorem 4.3 w.r.t. previous results and refer to Tables 1 and 2 in Appendix A.2 for a detailed comparison.

In terms of iteration complexity, Theorem 4.3 recovers the best known convergence rates in the tabular setting [Xiao, 2022], for both non-decreasing and exponentially increasing step-size schedules. While considering a more general setting, Theorem 4.3 matches or improves upon the convergence rate of previous works on policy gradient methods for non-tabular policy parameterizations that consider constant step-size schedules [Liu et al., 2019, Shani et al., 2020, Liu et al., 2020, Wang et al., 2020, Agarwal et al., 2021, Vaswani et al., 2022, Cayci et al., 2022] and matches the convergence speed of previous works that employ NPG, log-linear policies, and geometrically increasing step-size schedules [Alfano and Rebeschini, 2022, Yuan et al., 2023].

In terms of generality, the results in Theorem 4.3 hold without need to implement regularization [Cen et al., 2021, Zhan et al., 2021, Cayci et al., 2021, 2022, Lan, 2022], to impose bounded updates or smoothness of the policy [Agarwal et al., 2021, Liu et al., 2020], or to restrict the analysis to the case where the mirror map h is the negative entropy [Liu et al., 2019, Hu et al., 2022]. In particular, we improve upon the latest results for PMD with general policy parameterization by Vaswani et al. [2022], which only allow bounded step-sizes, where the bound can be particularly small, e.g., $(1-\gamma)^3/(2\gamma|\mathcal{A}|)$, and can slow down the learning process.

When $\mathcal{S}$ is a finite state space, a sufficient condition for ν_μ in (A3) to be bounded is requiring μ to have full support on $\mathcal{S}$. When μ does not have full support, one can still obtain a linear convergence rate for $V^\star(\mu') - V^T(\mu')$, for an arbitrary state distribution μ' with full support, and relate this quantity to $V^\star(\mu) - V^T(\mu)$. We refer to Appendix F for a more detailed discussion on the distribution mismatch coefficient.

An interpretation of our theory can be provided by connecting AMPO to the Policy Iteration algorithm (PI), which shares the linear convergence of AMPO. To see this, we first rewrite the Bregman projection step of AMPO (Line 2 of Algorithm 1) as

$$\begin{aligned}\pi_s^{t+1} &= \operatorname*{argmin}_{\pi\in\Delta(\mathcal{A})} \langle -\eta_t f_s^{\theta^{t+1}} + \nabla h(\pi_s^t), \pi\rangle + \mathcal{D}_h(\pi, \pi_s^t), \quad \forall s\in\mathcal{S},\\ &= \operatorname*{argmin}_{\pi\in\Delta(\mathcal{A})} \langle -f_s^{\theta^{t+1}} + \frac{1}{\eta_t}\nabla h(\pi_s^t), \pi\rangle + \frac{1}{\eta_t}\mathcal{D}_h(\pi, \pi_s^t), \quad \forall s\in\mathcal{S}.\end{aligned}$$

We refer to Appendix D.1 for a proof. Secondly, solving Line 1 of Algorithm 1 leads to $f_s^{t+1} - \frac{1}{\eta_t}\nabla h(\pi_s^t) \approx Q_s^{\pi^t}$. When the step-size $\eta_t \to \infty$, and $1/\eta_t \to 0$, the above viewpoint of the AMPO policy update thus becomes

$$\begin{aligned}\pi_s^{t+1} &= \operatorname*{argmin}_{\pi\in\Delta(\mathcal{A})} \langle -Q_s^{\pi^t}, \pi\rangle, \quad \forall s\in\mathcal{S},\\ &= \operatorname*{argmax}_{\pi\in\Delta(\mathcal{A})} \langle Q_s^{\pi^t}, \pi\rangle, \quad \forall s\in\mathcal{S},\end{aligned}$$

which is the PI algorithm. Here we ignore the Bregman divergence term $\mathcal{D}_h(\pi, \pi_s^t)$, as it is multiplied by $1/\eta_t$ that goes to 0. So AMPO behaves more and more like PI with a large enough step-size and is able to converge linearly as PI does.

4.2 Sample complexity for neural network parameterization

Neural networks are widely used in RL due to their empirical success in applications [Mnih et al., 2013, 2015, Silver et al., 2017]. However, few theoretical guarantees exist for using this parameterization class in policy optimization [Liu et al., 2019, Wang et al., 2020, Cayci et al., 2022]. Here, we show how we can use our framework and Theorem 4.3 to fill this gap by deriving a sample complexity result for AMPO when using neural network parameterization. We will consider the case where the parameterization class $\mathcal{F}^\Theta$ from Definition 3.1 belongs to the family of shallow ReLU networks, which have been shown to be universal approximators

[Jacot et al., 2018, Allen-Zhu et al., 2019b, Du et al., 2019b, Ji et al., 2019]. That is, for $(s,a) \in (\mathcal{S} \times \mathcal{A}) \subseteq \mathbb{R}^d$, define $f^\theta(s,a) = c^\top \sigma(W(s,a)+b)$, where $\sigma(y) = \max(y,0)$ for all $y \in \mathbb{R}$ is the ReLU activation function and is applied element-wisely, $c \in \mathbb{R}^m$, $W \in \mathbb{R}^{m \times d}$ and $b \in \mathbb{R}^m$.

At each iteration t of AMPO, we set $v^t = d_\mu^t$ and solve the regression problem in Line 1 of Algorithm 1 through stochastic gradient descent (SGD). In particular, we initialize entry-wise W_0 and b as i.i.d. random Gaussians from $\mathcal{N}(0, 1/m)$, and c as i.i.d. random Gaussians from $\mathcal{N}(0, \epsilon_A)$ with $\epsilon_A \in (0,1]$. Assuming access to a simulator with starting state-action distribution v^t, we run SGD for K steps on the matrix W, that is, for $k = 0, \dots, K-1$,

$$W_{k+1} = W_k - \alpha\big(f^{(k)}(s,a) - \widehat{Q}^t(s,a) - \eta_t^{-1}\nabla h(\pi_s^t)\big)\nabla_W f^{(k)}(s,a), \tag{17}$$

where $f^{(k)}(s,a) = c^\top \sigma((W_0 + W_k)(s,a) + b)$, $(s,a) \sim v_t$ and $\widehat{Q}^t(s,a)$ is an unbiased estimate of $Q^t(s,a)$ obtained through Algorithm 4. We can then present our result on the sample complexity of AMPO for neural network parameterization, which is based on our convergence Theorem 4.3 and an analysis of neural networks by Allen-Zhu et al. [2019a, Theorem 1].

Corollary 4.4. *In the setting of Theorem 4.3, let the parameterization class $\mathcal{F}^\Theta$ consist of sufficiently wide shallow ReLU neural networks. Using an exponentially increasing step-size and solving the minimization problem in Line 1 with SGD as in (17), the number of samples required by AMPO to find an ε-optimal policy with high probability is $\widetilde{\mathcal{O}}(C_v^2 \nu_\mu^5 / \varepsilon^4 (1-\gamma)^6)$, where ε has to be larger than a fixed and non-vanishing error floor.*

We provide a proof for Corollary 4.4 and an expression for the error floor in Appendix G. Note that in the case considered here, the sample complexity might be impacted by an additional $\texttt{poly}(\varepsilon^{-1})$ term. We refer to Appendix G for more details and a derivation which does not include an additional $\texttt{poly}(\varepsilon^{-1})$ term, enabling comparison with prior works.

5 Conclusion

We have introduced a novel framework for RL which, given a mirror map and any parameterization class, induces a policy class and an update rule. We have proven that this framework enjoys sublinear and linear convergence for non-decreasing and geometrically increasing step-size schedules, respectively. Future venues of investigation include studying the sample complexity of AMPO in on-policy and off-policy settings other than neural network parameterization, exploiting the properties of specific mirror maps to take advantage of the structure of the MDP and efficiently including representation learning in the algorithm. We refer to Appendix A.5 for a thorough discussion of future work. We believe that the main contribution of AMPO is to provide a general framework with theoretical guarantees that can help the analysis of specific algorithms and MDP structures. AMPO recovers and improves several convergence rate guarantees in the literature, but it is important to keep in consideration how previous works have exploited particular settings, while AMPO tackles the most general case. It will be interesting to see whether these previous works combined with our fast linear convergence result can derive new efficient sample complexity results.

References

Alekh Agarwal, Sham M. Kakade, Jason D. Lee, and Gaurav Mahajan. Optimality and approximation with policy gradient methods in markov decision processes. *Conference on Learning Theory*, 2020. (Cited on pages 7 and 23.)

Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *Journal of Machine Learning Research*, 2021. (Cited on pages 4, 5, 8, 9, 19, 20, 21, 22, 23, 35, and 37.)

Carlo Alfano and Patrick Rebeschini. Linear convergence for natural policy gradient with log-linear policy parametrization. *arXiv preprint arXiv:2209.15382*, 2022. (Cited on pages 2, 9, 18, 19, 20, 22, and 35.)

Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. *Advances in Neural Information Processing Systems*, 2019a. (Cited on pages 5, 10, 36, and 37.)

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, 2019b. (Cited on pages 10 and 39.)

Shun-ichi Amari. Natural gradient works efficiently in learning. *Neural Computation*, 1998. (Cited on page 19.)

Jonathan Baxter and Peter L. Bartlett. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, 2001. (Cited on page 1.)

Amir Beck. *First-Order Methods in Optimization*. SIAM-Society for Industrial and Applied Mathematics, 2017. (Cited on pages 4 and 23.)

Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 2003. (Cited on pages 19 and 24.)

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemyslaw Debiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019. (Cited on page 1.)

Jalaj Bhandari and Daniel Russo. Global optimality guarantees for policy gradient methods. *arXiv preprint arXiv:1906.01786*, 2019. (Cited on page 20.)

Jalaj Bhandari and Daniel Russo. On the linear convergence of policy gradient methods for finite MDPs. In *International Conference on Artificial Intelligence and Statistics*, 2021. (Cited on pages 20 and 22.)

Shalabh Bhatnagar, Richard S. Sutton, Mohammad Ghavamzadeh, and Mark Lee. Natural actor-critic algorithms. *Automatica*, 2009. (Cited on pages 1 and 19.)

Lev M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 1967. (Cited on page 3.)

Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 2015. (Cited on pages 3, 24, 28, and 29.)

Semih Cayci, Niao He, and R Srikant. Linear convergence of entropy-regularized natural policy gradient with linear function approximation. *arXiv preprint arXiv:2106.04096*, 2021. (Cited on pages 2, 8, 9, 20, 22, 23, and 35.)

Semih Cayci, Niao He, and R Srikant. Finite-time analysis of entropy-regularized neural natural actor-critic algorithm. *arXiv preprint arXiv:2206.00833*, 2022. (Cited on pages 9, 18, 20, 21, 35, 39, and 40.)

Shicong Cen, Chen Cheng, Yuxin Chen, Yuting Wei, and Yuejie Chi. Fast global convergence of natural policy gradient methods with entropy regularization. *Operations Research*, 2021. (Cited on pages 2, 9, 20, 22, and 35.)

Gong Chen and Marc Teboulle. Convergence analysis of a proximal-like minimization algorithm using bregman functions. *SIAM Journal on Optimization*, 1993. (Cited on pages 2 and 28.)

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*, 2022. (Cited on pages 2, 20, 22, and 35.)

Ashok Cutkosky and Francesco Orabona. Momentum-based variance reduction in non-convex sgd. In *Advances in Neural Information Processing Systems*, 2019. (Cited on pages 20 and 23.)

Yuhao Ding, Junzi Zhang, and Javad Lavaei. On the global optimum convergence of momentum-based policy gradient. In *International Conference on Artificial Intelligence and Statistics*, 2022. (Cited on page 20.)

Simon Du, Akshay Krishnamurthy, Nan Jiang, Alekh Agarwal, Miroslav Dudik, and John Langford. Provably efficient rl with rich observations via latent state decoding. In *International Conference on Machine Learning*, 2019a. (Cited on page 23.)

Simon Du, Sham Kakade, Jason Lee, Shachar Lovett, Gaurav Mahajan, Wen Sun, and Ruosong Wang. Bilinear classes: A structural framework for provable generalization in RL. In *International Conference on Machine Learning*, 2021. (Cited on page 23.)

Simon S. Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations*, 2019b. (Cited on page 10.)

Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *International conference on machine learning*, 2018. (Cited on page 7.)

Ilyas Fatkhullin, Jalal Etesami, Niao He, and Negar Kiyavash. Sharp analysis of stochastic optimization under global Kurdyka-łojasiewicz inequality. In *Advances in Neural Information Processing Systems*, 2022. (Cited on page 20.)

Ilyas Fatkhullin, Anas Barakat, Anastasia Kireeva, and Niao He. Stochastic policy gradient methods: Improved sample complexity for fisher-non-degenerate policies. *arXiv preprint arXiv:2302.01734*, 2023. (Cited on page 20.)

Maryam Fazel, Rong Ge, Sham Kakade, and Mehran Mesbahi. Global convergence of policy gradient methods for the linear quadratic regulator. In *International Conference on Machine Learning*, 2018. (Cited on page 20.)

Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized Markov decision processes. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2160–2169. PMLR, 09–15 Jun 2019. (Cited on page 24.)

Udaya Ghai, Elad Hazan, and Yoram Singer. Exponentiated gradient meets gradient descent. In *International Conference on Algorithmic Learning Theory*, 2020. (Cited on pages 7 and 28.)

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016. (Cited on page 5.)

Yuzheng Hu, Ziwei Ji, and Matus Telgarsky. Actor-critic is implicitly biased towards high entropy optimal policies. In *International Conference on Learning Representations*, 2022. (Cited on pages 7, 9, 18, 20, and 21.)

Feihu Huang, Shangqian Gao, and Heng Huang. Bregman gradient policy optimization. In *International Conference on Learning Representations*, 2022. (Cited on pages 20 and 21.)

Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, 2018. (Cited on page 10.)

Ziwei Ji, Matus Telgarsky, and Ruicheng Xian. Neural tangent kernels, transportation mappings, and universal approximation. In *International Conference on Learning Representations*, 2019. (Cited on pages 10, 34, 39, and 40.)

Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, 2020. (Cited on pages 20 and 23.)

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning*, 2002. (Cited on page 31.)

Sham M. Kakade. A natural policy gradient. *Advances in Neural Information Processing Systems*, 2002. (Cited on pages 1, 5, and 19.)

William Karush. Minima of functions of several variables with inequalities as side conditions. Master's thesis, Department of Mathematics, University of Chicago, Chicago, IL, USA, 1939. (Cited on page 25.)

Michael J. Kearns and Daphne Koller. Efficient reinforcement learning in factored mdps. In *International Joint Conference on Artificial Intelligence*, 1999. (Cited on page 23.)

Sajad Khodadadian, Zaiwei Chen, and Siva Theja Maguluri. Finite-sample analysis of off-policy natural actor-critic algorithm. In *International Conference on Machine Learning*, 2021a. (Cited on pages 19 and 21.)

Sajad Khodadadian, Prakirt Raj Jhunjhunwala, Sushil Mahavir Varma, and Siva Theja Maguluri. On the linear convergence of natural policy gradient algorithm. In *IEEE Conference on Decision and Control*, 2021b. (Cited on pages 20 and 22.)

Sajad Khodadadian, Prakirt Raj Jhunjhunwala, Sushil Mahavir Varma, and Siva Theja Maguluri. On linear and super-linear convergence of natural policy gradient algorithm. *Systems and Control Letters*, 2022. (Cited on pages 20 and 22.)

Vijay Konda and John Tsitsiklis. Actor-critic algorithms. In *Advances in Neural Information Processing Systems*, 2000. (Cited on page 1.)

Walid Krichene, Syrine Krichene, and Alexandre Bayen. Efficient bregman projections onto the simplex. In *IEEE Conference on Decision and Control*, 2015. (Cited on pages 6, 7, 25, 26, and 27.)

Jakub Grudzien Kuba, Christian A Schroeder De Witt, and Jakob Foerster. Mirror learning: A unifying framework of policy optimisation. In *International Conference on Machine Learning*, 2022. (Cited on pages 2, 18, and 20.)

Harold W. Kuhn and Albert W. Tucker. Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, 1951. (Cited on page 25.)

Solomon Kullback and Richard A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 1951. (Cited on page 1.)

Guanghui Lan. Policy mirror descent for reinforcement learning: Linear convergence, new sampling complexity, and generalized problem classes. *Mathematical programming*, 2022. (Cited on pages 1, 2, 9, 19, 20, 22, and 35.)

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020. (Cited on page 24.)

Yan Li and Guanghui Lan. Policy mirror descent inherently explores action space. *arXiv preprint arXiv:2303.04386*, 2023. (Cited on page 7.)

Yan Li, Tuo Zhao, and Guanghui Lan. Homotopic policy mirror descent: Policy convergence, implicit regularization, and improved sample complexity. *arXiv preprint arXiv:2201.09457*, 2022. (Cited on pages 20 and 22.)

Zhize Li, Hongyan Bao, Xiangliang Zhang, and Peter Richtarik. PAGE: A simple and optimal probabilistic gradient estimator for nonconvex optimization. In *International Conference on Machine Learning*, 2021. (Cited on pages 20 and 23.)

Boyi Liu, Qi Cai, Zhuoran Yang, and Zhaoran Wang. Neural trust region/proximal policy optimization attains globally optimal policy. *Advances in Neural Information Processing Systems*, 2019. (Cited on pages 7, 9, 20, and 21.)

Yanli Liu, Kaiqing Zhang, Tamer Basar, and Wotao Yin. An improved analysis of (variance-reduced) policy gradient and natural policy gradient methods. *Advances in Neural Information Processing Systems*, 2020. (Cited on pages 9, 20, 21, and 23.)

Stanislaw Lojasiewicz. Une propriété topologique des sous-ensembles analytiques réels. *Les équations aux dérivées partielles*, 1963. (Cited on page 20.)

Saeed Masiha, Saber Salehkaleybar, Niao He, Negar Kiyavash, and Patrick Thiran. Stochastic second-order methods improve best-known sample complexity of SGD for gradient-dominated functions. In *Advances in Neural Information Processing Systems*, 2022. (Cited on page 20.)

Jincheng Mei, Chenjun Xiao, Csaba Szepesvari, and Dale Schuurmans. On the global convergence rates of softmax policy gradient methods. In *International Conference on Machine Learning*, 2020. (Cited on page 20.)

Jincheng Mei, Yue Gao, Bo Dai, Csaba Szepesvari, and Dale Schuurmans. Leveraging non-uniformity in first-order non-convex optimization. In *International Conference on Machine Learning*, 2021. (Cited on page 20.)

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. (Cited on page 9.)

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 2015. (Cited on pages 8 and 9.)

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, 2016. (Cited on page 1.)

Arkadi Nemirovski and David B. Yudin. *Problem Complexity and Method Efficiency in Optimization*. Wiley Interscience, 1983. (Cited on pages 2, 3, and 19.)

Yurii E. Nesterov and Boris T. Polyak. Cubic regularization of Newton method and its global performance. *Mathematical Programming*, 2006. (Cited on page 20.)

Gergely Neu, Anders Jonsson, and Vicenç Gómez. A unified view of entropy-regularized markov decision processes. *arXiv preprint arXiv:1705.07798*, 2017. (Cited on page 19.)

Lam M. Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. SARAH: A novel method for machine learning problems using stochastic recursive gradient. In *International Conference on Machine Learning*, 2017. (Cited on pages 20 and 23.)

Francesco Orabona. A modern introduction to online learning, 2020. URL https://open.bu.edu/handle/2144/40900. (Cited on pages 7 and 27.)

Jan Peters and Stefan Schaal. Natural actor-critic. *Neurocomputing*, 2008. (Cited on pages 1 and 19.)

Boris T. Polyak. Gradient methods for the minimisation of functionals. *USSR Computational Mathematics and Mathematical Physics*, 1963. (Cited on page 20.)

Bruno Scherrer. Approximate policy iteration schemes: A comparison. In *International Conference on Machine Learning*, 2014. (Cited on page 36.)

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, 2015a. (Cited on pages 1 and 17.)

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015b. (Cited on page 7.)

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. (Cited on pages 1 and 17.)

Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. Safe, multi-agent, reinforcement learning for autonomous driving. *arXiv preprint arXiv:1610.03295*, 2016. (Cited on page 1.)

Lior Shani, Yonathan Efroni, and Shie Mannor. Adaptive trust region policy optimization: Global convergence and faster rates for regularized MDPs. In *AAAI Conference on Artificial Intelligence*, 2020. (Cited on pages 2, 6, 9, 19, and 21.)

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of Go without human knowledge. *Nature*, 2017. (Cited on page 9.)

Wen Sun, Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Model-based rl in contextual decision processes: Pac bounds and exponential improvements over model-free approaches. In *Conference on Learning Theory*, 2019. (Cited on page 23.)

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018. (Cited on page 17.)

Richard S Sutton, Andrew G Barto, et al. *Introduction to reinforcement learning*. MIT press Cambridge, 1998. (Cited on page 7.)

Richard S. Sutton, David A. McAllester, Satinder P. Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, 1999. (Cited on pages 1, 3, and 5.)

Manan Tomar, Lior Shani, Yonathan Efroni, and Mohammad Ghavamzadeh. Mirror descent policy optimization. In *International Conference on Learning Representations*, 2022. (Cited on pages 2, 5, 7, and 17.)

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. In *International Conference on Artificial Intelligence and Statistics*, 2022. (Cited on pages 2, 5, 7, 9, 18, and 20.)

Lingxiao Wang, Qi Cai, Zhuoran Yang, and Zhaoran Wang. Neural policy gradient methods: Global optimality and rates of convergence. In *International Conference on Learning Representations*, 2020. (Cited on pages 8, 9, 20, and 21.)

Weiran Wang and Miguel A Carreira-Perpinán. Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application. *arXiv preprint arXiv:1309.1541*, 2013. (Cited on page 26.)

Ronald J. Williams and Jing Peng. Function optimization using connectionist reinforcement learning algorithms. *Connection Science*, 1991. (Cited on page 1.)

Lin Xiao. On the convergence rates of policy gradient methods. *Journal of Machine Learning Research*, 2022. (Cited on pages 2, 5, 6, 7, 8, 9, 18, 19, 20, 21, 22, 26, 28, 30, 31, 35, and 36.)

Tengyu Xu, Zhe Wang, and Yingbin Liang. Improving sample complexity bounds for (natural) actor-critic algorithms. In *Advances in Neural Information Processing Systems*, 2020. (Cited on pages 20 and 21.)

Long Yang, Yu Zhang, Gang Zheng, Qian Zheng, Pengfei Li, Jianhang Huang, and Gang Pan. Policy optimization with stochastic mirror descent. *AAAI Conference on Artificial Intelligence*, 2022. (Cited on pages 20 and 21.)

Rui Yuan, Robert M. Gower, and Alessandro Lazaric. A general sample complexity analysis of vanilla policy gradient. In *International Conference on Artificial Intelligence and Statistics*, 2022. (Cited on pages 20 and 23.)

Rui Yuan, Simon Shaolei Du, Robert M. Gower, Alessandro Lazaric, and Lin Xiao. Linear convergence of natural policy gradient methods with log-linear policies. In *International Conference on Learning Representations*, 2023. (Cited on pages 2, 6, 7, 8, 9, 18, 19, 20, 21, 22, 35, and 36.)

Andrea Zanette, Ching-An Cheng, and Alekh Agarwal. Cautiously optimistic policy optimization and exploration with linear function approximation. In *Conference on Learning Theory*, 2021. (Cited on pages 18, 20, and 21.)

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. *arXiv preprint arXiv:2105.11066*, 2021. (Cited on pages 2, 9, 20, 22, and 35.)

Junyu Zhang, Alec Koppel, Amrit Singh Bedi, Csaba Szepesvari, and Mengdi Wang. Variational policy gradient method for reinforcement learning with general utilities. In *Advances in Neural Information Processing Systems*, 2020. (Cited on pages 8 and 20.)

Junyu Zhang, Chengzhuo Ni, Zheng Yu, Csaba Szepesvari, and Mengdi Wang. On the convergence and sample efficiency of variance-reduced policy gradient method. In *Advances in Neural Information Processing Systems*, 2021. (Cited on page 20.)

Appendix

Here we provide the related work discussion, the deferred proofs from the main paper and some additional noteworthy observations.

A Related work

We provide an extended discussion for the context of our work, including a comparison of different PMD frameworks and a comparison of the convergence theories of PMD in the literature. Furthermore, we discuss future work, such as extending our analysis to the dual averaging updates and developing sample complexity analysis of AMPO.

A.1 Comparisons with other policy optimization frameworks

In this section, we give a comparison with some of the most popular policy optimization algorithms in the literature.

Generalised Policy Iteration [Sutton and Barto, 2018]. The update consists in evaluating the Q-function of the policy and obtaining the new policy by acting greedily with respect to the estimated Q-function. That is, for all $s \in \mathcal{S}$,

$$\pi_s^{t+1} \in \operatorname*{argmax}_{\pi_s \in \Delta(\mathcal{A})} \langle Q_s^t, \pi_s \rangle. \tag{18}$$

AMPO recovers this algorithm when we have access to the value of Q_s^t and $\eta_t \to +\infty$ for all times t.

Trust Region Policy Optimization [Schulman et al., 2015a]. The TRPO update is as follows:

$$\pi^{t+1} \in \operatorname*{argmax}_{\pi \in \Pi} \mathbb{E}_{s \sim d_\mu^t} \left[\langle A_s^t, \pi_s \rangle \right], \tag{19}$$
$$\text{such that} \quad \mathbb{E}_{s \sim d_\mu^t} \left[D_h(\pi_s^t, \pi_s) \right] \leq \delta,$$

where $A_s^t = Q_s^t - V^t$ represents the advantage function, h is the negative entropy and $\delta > 0$. TRPO is equivalent to AMPO when at each time t, the admissible policy class is $\Pi^t = \{\pi \in \Delta(\mathcal{A})^{\mathcal{S}} : \mathbb{E}_{s \sim d_\mu^t} D_h(\pi_s^t, \pi_s) \leq \delta\}$, we have access to the value of Q_s^t and $\eta_t \to +\infty$.

Proximal Policy Optimization [Schulman et al., 2017]. The Proximal Policy Optimization (PPO) update consists in maximizing a surrogate function depending on the policy gradient with respect to the new policy. Namely,

$$\pi^{t+1} \in \operatorname*{argmax}_{\pi \in \Pi} \mathbb{E}_{s \sim d_\mu^t} \left[L(\pi_s, \pi_s^t) \right], \tag{20}$$

with

$$L(\pi_s, \pi_s^t) = \mathbb{E}_{a \sim \pi^t}[\min\left(r^\pi(s,a) A^t(s,a), \operatorname{clip}(r^\pi(s,a), 1 \pm \epsilon) A^t(s,a)\right)],$$

where $r^\pi(s,a) = \pi(s,a)/\pi^t(s,a)$ is the probability ratio between the current policy π^t and the new one, and the function $\operatorname{clip}(r^\pi(s,a), 1 \pm \epsilon)$ clips the probability ratio $r^\pi(s,a)$ to be no more than $1 + \epsilon$ and no less than $1 - \epsilon$. PPO has also a KL variation [Schulman et al., 2017, Section 4], where the objective function L is defined as

$$L(\pi_s, \pi_s^t) = \eta_t \langle A_s^t, \pi_s \rangle - D_h(\pi_s^t, \pi_s),$$

where h is the negative entropy. In an exact setting and when $\Pi = \Delta(\mathcal{A})^{\mathcal{S}}$, the KL variation of PPO differs from AMPO because it inverts the terms in the Bregman divergence penalty.

Mirror Descent Policy Optimization [Tomar et al., 2022]. The algorithm consists in the following update:

$$\pi^{t+1} \in \operatorname*{argmax}_{\pi \in \Pi} \mathbb{E}_{s \sim d_\mu^t}[\langle A_s^t, \pi_s \rangle - D_h(\pi_s, \pi_s^t)], \tag{21}$$

where Π is a parameterized policy class. While it is equivalent to AMPO in an exact setting and when $\Pi = \Delta(\mathcal{A})^{\mathcal{S}}$, as we show in Appendix B, the difference between the two algorithms lies on the approximation of the exact algorithm.

Functional Mirror Ascent Policy Gradient [Vaswani et al., 2022]. The algorithm consists in the following update:

$$\pi^{t+1} \in \underset{\pi^\theta:\, \theta\in\Theta}{\operatorname{argmax}} \mathbb{E}_{s\sim d_\mu^t}[V^{\pi^t}(\mu) + \langle \nabla_{\pi_s} V^{\pi^t}(\mu)\big|_{\pi=\pi^t}, \pi_s^\theta - \pi_s^t\rangle - D_h(\pi_s^\theta, \pi_s^t)] \tag{22}$$
$$= \underset{\pi^\theta,\, \theta\in\Theta}{\operatorname{argmax}} \mathbb{E}_{s\sim d_\mu^t}[\langle Q_s^t, \pi_s^\theta\rangle - D_h(\pi_s^\theta, \pi_s^t)],$$

The second line is obtained by the definition of V^t and the policy gradient theorem (3). The discussion is the same as the previous algorithm.

Mirror Learning [Kuba et al., 2022]. The on-policy version of the algorithm consists in the following update:

$$\pi^{t+1} = \underset{\pi\in\Pi(\pi^t)}{\operatorname{argmax}} \mathbb{E}_{s\sim d_\mu^t}[\langle Q_s^t, \pi_s\rangle - D(\pi_s, \pi_s^t)], \tag{23}$$

where $\Pi(\pi^t)$ is a policy class that depends on the current policy π^t and the drift functional D is defined as a map $D : \Delta(\mathcal{A})\times\Delta(\mathcal{A}) \to \mathbb{R}$ such that $D(\pi_s, \bar{\pi}_s) \geq 0$ and $\nabla_{\pi_s} D(\pi_s, \bar{\pi}_s)\big|_{\pi_s=\bar{\pi}_s} = 0$. The drift functional D recovers the Bregman divergence as a particular case, in which case Mirror Learning is equivalent to AMPO in an exact setting and when $\Pi = \Delta(\mathcal{A})^{\mathcal{S}}$. Once more, the main difference between the two algorithms lies on the approximation of the exact algorithm.

A.2 Discussion on related work

Our Contributions. Our work provides a framework for policy optimization – AMPO. For AMPO, we establish in Theorem 4.3 both $\mathcal{O}(1/T)$ convergence guarantee by using a non-decreasing step-size and linear convergence guarantee by using a geometrically increasing step-size. Our contributions, with respect to previous literature on sublinear and linear convergence of policy optimization methods, can be summarized as follows.

- The generalities of our framework and of Lemma 4.1 allow Theorem 4.3 to unify previous results in the literature and generate new theoretically sound algorithms under one guise. Indeed, both the sublinear and the linear convergence analysis of natural policy gradient (NPG) with softmax tabular policies [Xiao, 2022] or with log-linear policies [Alfano and Rebeschini, 2022, Yuan et al., 2023] are special cases of our general analysis. Thus, Theorem 4.3 recovers the best known convergence rates in both the tabular setting [Xiao, 2022] and the non-tabular setting [Cayci et al., 2022, Alfano and Rebeschini, 2022, Yuan et al., 2023]. AMPO also generates new algorithms by selecting mirror maps, e.g., ϵ-negative entropy mirror map in Appendix C.2 associated with Algorithm 2, and generalizes the projected Q-descent algorithm [Xiao, 2022] from the tabular setting to a general parameterization class $\mathcal{F}^\Theta$.
- As discussed in Section 4.1, the results in Theorem 4.3 hold for a general setting with less restrictions than previous works. The generality of the assumptions of Theorem 4.3 allows the application of our theory to specific settings, where existing sample complexity analyses could be improved thanks to the linear convergence of AMPO. For instance, since Theorem 4.3 holds for any structural MDP, AMPO could be directly applied to the linear MDP setting to derive a sample complexity analysis of AMPO which may improve that of Zanette et al. [2021] and Hu et al. [2022]. As we discuss in Appendix A.5, this is a promising direction for future work.
- From a technical point of view, our main contributions are: Definition 3.1 introduces a novel method of incorporating general parameterization in the policy; the update in Line 1 of Algorithm 1 simplifies the policy optimization step into a regression problem; and Lemma 4.1 establishes a key result for policies belonging to the class in Definition 3.1. Together, these innovations have allowed us to establish new state-of-the-art results in Theorem 4.3.

In particular, our technical novelty with respect to Xiao [2022], Alfano and Rebeschini [2022], and Yuan et al. [2023] can be summarized as follows.

- As to the algorithm design, AMPO represents an innovation. The PMD algorithm proposed by Xiao [2022] is strictly constrained to the tabular setting and, while it is well defined for any mirror map, it cannot include general parameterization. Alfano and Rebeschini [2022] and Yuan et al. [2023] propose a first generalization of the PMD algorithm, but are limited to consider log-linear parameterization and the entropy mirror map. On the contrary, AMPO solves the problem of incorporating general parameterization in the policy thanks to Definition 3.1 and the policy update in Line 1 of Algorithm 1. This innovation is key for the generality of the algorithm, as it allows AMPO to employ any mirror map and any parameterization class. Moreover, AMPO is computationally efficient for a large class of mirror maps (see Appendix C.2). We expect our design to be useful in deep RL, where the policy is usually parameterized by a neural network whose last layer is a softmax transformation. Our policy definition can be implemented in this setting by replacing the softmax layer with a Bregman projection.

- As to the assumptions necessary for convergence guarantees, we have weaker assumptions. Xiao [2022] requires an L_∞ supremum norm bound on the approximation error of Q^t, i.e., $\|\widehat{Q}^t - Q^t\|_\infty \leq \varepsilon_{\text{approx}}$, for all $t \leq T$. Alfano and Rebeschini [2022] and Yuan et al. [2023] require an L_2-norm bound on the error of the linear approximation of Q^t, i.e., $\|w^\top \phi - Q^t\|_\infty \leq \varepsilon_{\text{approx}}$ for some feature function $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ and vector $w \in \mathbb{R}^d$, for all $t \leq T$. Our approximation error assumption (A1) represents an improvement as it does not constrain the bound to hold in L_∞-norm or involve linear function approximation. On the contrary, Assumption (A1) allows any regression model, in particular neural networks. We further relax Assumption (A1) in Appendix E and show that the approximation error bound can be larger for earlier iterations. Additionally, we improve upon the concentrability coefficients of Yuan et al. [2023] as we define them with respect to an arbitrary distribution.

- As to the analysis of the algorithm, while we borrow tools from Xiao [2022], our results are not simple extensions. In fact, it is not clear from Xiao [2022], Alfano and Rebeschini [2022], and Yuan et al. [2023] whether PMD could have theoretical guarantees in a setting with general parameterization and an arbitrary mirror map. The two main problems on this front are the non-convexity of the policy class, which prevents using the three-point descent lemma by Xiao [2022, Lemma 6], and the fact that the three-point identity by Alfano and Rebeschini [2022, Equation 4] only holds for the negative entropy mirror map. Our Lemma 4.1 is the first result that successfully addresses general policy parameterization and arbitrary mirror maps. Additionally, we provide a sample complexity analysis of AMPO when employing shallow neural networks that improves upon previous state-of-the-art results in this setting. We further improve this sample complexity analysis in Appendix G, where we consider an approximation assumption weaker than Assumption (A1) (see Appendix E).

Related literature. Lately, the impressive empirical success of policy gradient (PG) - type methods has catalyzed the development of theoretically sound gradient-based algorithms for policy optimization. In particular, there has been a lot of attention around algorithms inspired by mirror descent (MD) [Nemirovski and Yudin, 1983, Beck and Teboulle, 2003] and, more specifically, by natural gradient descent [Amari, 1998]. These two approaches led to policy mirror descent (PMD) methods [Shani et al., 2020, Lan, 2022] and natural policy gradient (NPG) methods [Kakade, 2002], which as first shown by Neu et al. [2017] is a particular case of PMD. Leveraging different techniques from the MD literature, it has been established that PMD, NPG and their variants converge to the global optimum in different settings. We refer to global optimum convergence as an analysis that guarantees that $V^\star(\mu) - \mathbb{E}\left[V^T(\mu)\right] \leq \epsilon$ after T iterations with $\epsilon > 0$. As an important variant of NPG, we will also discuss the literature of the convergence analysis of natural actor-critic (NAC) [Peters and Schaal, 2008, Bhatnagar et al., 2009].

Sublinear convergence analyses of PMD, NPG and NAC. For softmax tabular policies, Shani et al. [2020] establish a $\mathcal{O}(1/\sqrt{T})$ convergence rate for unregularized NPG and $\mathcal{O}(1/T)$ for regularized NPG. Agarwal et al. [2021], Khodadadian et al. [2021a] and Xiao

[2022] improve the convergence rate for unregularized NPG and NAC to $\mathcal{O}(1/T)$ and Xiao [2022] extends the same convergence rate to projected Q-descent.

In the function approximation regime, Zanette et al. [2021] and Hu et al. [2022] achieve $\mathcal{O}(1/\sqrt{T})$ convergence rate by developing variants of PMD methods for the linear MDP [Jin et al., 2020] setting. The same $\mathcal{O}(1/\sqrt{T})$ convergence rate is obtained by Agarwal et al. [2021] for both log-linear and smooth policies, while Yuan et al. [2023] improve the convergence rate to $\mathcal{O}(1/T)$ for log-linear policies. For smooth policies, the convergence rate is later improved to $\mathcal{O}(1/T)$ either by adding an extra Fisher-non-degeneracy condition on the policies [Liu et al., 2020] or by analyzing NAC under Markovian sampling [Xu et al., 2020]. Yang et al. [2022] and Huang et al. [2022] consider Lipschitz and smooth policies [Yuan et al., 2022], obtain $\mathcal{O}(1/\sqrt{T})$ convergence rates for PMD-type methods and faster $\mathcal{O}(1/T)$ convergence rates by applying the variance reduction techniques SARAH [Nguyen et al., 2017] and STORM [Cutkosky and Orabona, 2019], respectively. As for neural policy parameterization, Liu et al. [2019] establish a $\mathcal{O}(1/\sqrt{T})$ convergence rate for two-layer neural PPO. The same $\mathcal{O}(1/\sqrt{T})$ convergence rate is established by Wang et al. [2020] for two-layer neural NAC, which is later improved to $\mathcal{O}(1/T)$ by Cayci et al. [2022], using entropy regularization.

We highlight that all the sublinear convergence analyses mentioned above, for both softmax tabular policies and the function approximation regime, are obtained by either using a decaying step-size or a constant step-size. We refer to Table 1 for an overview of recent sublinear convergence analyses of NPG/PMD.

Linear convergence analysis of PMD, NPG, NAC and other PG methods. In the softmax tabular policy settings, the linear convergence guarantees of NPG and PMD are achieved by either adding regularization [Cen et al., 2021, Zhan et al., 2021, Lan, 2022, Li et al., 2022] or by varying the step-sizes [Bhandari and Russo, 2021, Khodadadian et al., 2021b, 2022, Xiao, 2022].

In the function approximation regime, the linear convergence guarantees are achieved for NPG with log-linear policies, either by adding entropy regularization [Cayci et al., 2021] or by choosing geometrically increasing step-sizes [Alfano and Rebeschini, 2022, Yuan et al., 2023]. It can also be achieved for NAC with log-linear policy by using adaptive increasing step-sizes [Chen and Theja Maguluri, 2022].

We refer to Table 2 for an overview of recent linear convergence analyses of NPG/PMD.

Alternatively, by leveraging a Polyak-Lojasiewicz (PL) condition [Polyak, 1963, Lojasiewicz, 1963], fast linear convergence results can be achieved for PG methods under different settings, such as linear quadratic control problems [Fazel et al., 2018] and softmax tabular policies with entropy regularization [Mei et al., 2020, Yuan et al., 2022]. The PL condition is widely explored by Bhandari and Russo [2019] to identify more general MDP settings. Similar to the cases of NPG and PMD, by choosing adaptive step-sizes through exact line search [Bhandari and Russo, 2021] or by exploiting non-uniform smoothness [Mei et al., 2021], linear convergence of PG can also be obtained for the softmax tabular policy without regularization. When the PL condition is relaxed to other weaker conditions, PG methods combined with variance reduction methods such as SARAH [Nguyen et al., 2017] and PAGE [Li et al., 2021] can also achieve linear convergence. This is shown by Fatkhullin et al. [2022, 2023] when the PL condition is replaced by the weak PL condition [Yuan et al., 2022], which is satisfied by Fisher-non-degenerate policies [Ding et al., 2022]. It is also shown by Zhang et al. [2021], where the MDP satisfies some hidden convexity property which contains a similar property to the weak PL condition studied by Zhang et al. [2020]. Lastly, linear convergence is established for the cubic-regularized Newton method [Nesterov and Polyak, 2006], a second-order method, applied on Fisher-non-degenerate policies combined with variance reduction [Masiha et al., 2022].

Outside the literature focusing on finite time convergence guarantees, Vaswani et al. [2022] and Kuba et al. [2022] provide a theoretical analysis for variations of PMD, showing monotonic improvements for their frameworks. Additionally, Kuba et al. [2022] give an infinite time convergence guarantee for their framework.

Table 1: Overview of sublinear convergence results for NPG and PMD methods with constant step-size in different settings. The dark blue cells contain our new results. The light blue cells contain previously known results that we recover as particular cases of our analysis. The pink cells contain previously best known results upon which we improve by providing a faster convergence rate. White cells contain existing results that are already improved by other literature or that we could not recover under our general analysis.

Algorithm	**Rate**	**Comparisons to our works**
Setting: Softmax tabular policies		
Adaptive TRPO [Shani et al., 2020]	$\mathcal{O}(1/\sqrt{T})$	They employ regularization
Tabular off-policy NAC [Khodadadian et al., 2021a]	$\mathcal{O}(1/T)$	We have a weaker approximation error asm. with L_2 instead of L_∞
Tabular NPG [Agarwal et al., 2021]	$\mathcal{O}(1/T)$	
Tabular NPG/projected Q-descent [Xiao, 2022]	$\mathcal{O}(1/T)$	We recover their results when $f^\theta(s,a) = \theta_{s,a}$; we have a weaker approximation error asm. with L_2 instead of L_∞
Setting: Log-linear policies		
Q-NPG [Agarwal et al., 2021]	$\mathcal{O}(1/\sqrt{T})$	
Q-NPG/NPG [Yuan et al., 2023]	$\mathcal{O}(1/T)$	We recover their results when $f^\theta(s,a)$ is linear
Setting: Softmax two-layer neural policies		
Neural PPO [Liu et al., 2019]	$\mathcal{O}(1/\sqrt{T})$	
Neural NAC [Wang et al., 2020]	$\mathcal{O}(1/\sqrt{T})$	
Regularized neural NAC [Cayci et al., 2022]	$\mathcal{O}(1/T)$	They employ regularization
Setting: Linear MDP		
NPG [Zanette et al., 2021] [Hu et al., 2022]	$\mathcal{O}(1/\sqrt{T})$	
Setting: Smooth policies		
NPG [Agarwal et al., 2021]	$\mathcal{O}(1/\sqrt{T})$	
NAC under Markovian sampling [Xu et al., 2020]	$\mathcal{O}(1/T)$	
NPG with Fisher-non-degenerate policies [Liu et al., 2020]	$\mathcal{O}(1/T)$	
Setting: Lipschitz and Smooth policies		
Variance reduced PMD [Yang et al., 2022, Huang et al., 2022]	$\mathcal{O}(1/T)$	
Setting: Bregman projected policies with general parameterization and mirror map		
AMPO (Theorem 4.3, this work)	$\mathcal{O}(1/T)$	

Table 2: Overview of linear convergence results for NPG and PMD methods in different settings. The darker cells contain our new results. The light cells contain previously known results that we recover as special cases of our analysis, and extend the permitted concentrability coefficients settings. White cells contain existing results that we could not recover under our general analysis.

Algorithm	Reg.	C.S.	A.I.S.	N.I.S.*	Error assumption
Setting: Softmax tabular policies					
NPG [Cen et al., 2021]	✓	✓			L_∞
PMD [Zhan et al., 2021]	✓	✓			L_∞
NPG [Lan, 2022]	✓			✓	L_∞
NPG [Li et al., 2022]	✓			✓	L_∞
NPG [Bhandari and Russo, 2021]			✓		
NPG [Khodadadian et al., 2021b] [Khodadadian et al., 2022]			✓		L_∞
NPG / Projected Q-descent [Xiao, 2022]				✓	L_∞
Setting: Log-linear policies					
NPG [Cayci et al., 2021]	✓	✓			L_2
Off-policy NAC [Chen and Theja Maguluri, 2022]			✓		L_∞
Q-NPG [Alfano and Rebeschini, 2022]				✓	L_2
Q-NPG/NPG [Yuan et al., 2023]				✓	L_2
Setting: Bregman projected policies with general parameterization and mirror map					
AMPO (Theorem 4.3, this work)				✓	L_2

* **Reg.**: regularization; **C.S.**: constant step-size; **A.I.S.**: Adaptive increasing step-size; **N.I.S.**: Non-adaptive increasing step-size.

A.3 Comparison with previous compatible function approximation frameworks

Agarwal et al. [2021] study NPG with smooth policies through compatible function approximation and propose the following algorithm. Let $\{\pi^\theta : \theta \in \Theta\}$ be a policy class such that $\log \pi^\theta(a|s)$ is a β-smooth function of θ for all $s \in \mathcal{S}$, $a \in \mathcal{A}$. At each iteration t, update θ^t as

$$\theta^{t+1} = \theta^t + \eta w^t,$$

with

$$w^t \in \operatorname*{argmin}_{\|w\|_2 \leq W} \left\| A^t - w^\top \nabla_\theta \log \pi^t \right\|_{L_2(d_\mu^t \cdot \pi^t)}, \tag{24}$$

where $W > 0$ and $A_s^t = Q_s^t - V^t$ represents the advantage function. While both the algorithms proposed by Agarwal et al. [2021] and AMPO involve regression problems, the one in (24) is restricted to linearly approximate A^t with $\nabla_\theta \log \pi^t$, whereas the one in Line 1 of Algorithm 1 is widely relaxed to approximate A^t with an arbitrary class of functions $\mathcal{F}^\Theta$. Additionally, (24) depends on the distribution d_μ^t, while Line 1 of Algorithm 1 does not and allows off-policy updates involving an arbitrary distribution v^t. Thus our framework not only recovers the NPG setting in Agarwal et al. [2021], but also greatly improve the performance of the framework thanks to a much richer representation power of $\mathcal{F}^\Theta$ and the flexibility of choosing different distributions v^t in either an on-policy or an off-policy setting.

A.4 Benefits of the structure of AMPO compared to (11)

We provide here a setting where the optimization problem in Line 1 of Algorithm 1 inherits curvature from the parameterization function, i.e., strong convexity, while the optimization problem in (11) does not. Let $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^k$ be a feature vector and let the parameterization class be defined as $\mathcal{F}^\Theta = \{\theta^\top \phi : \theta \in \Theta\}$. For this setting and for any mirror map, the minimization problem in Line 1 of Algorithm 1 is convex, as

$$\nabla^2_\theta \left[\left\| \theta^\top \phi - Q^t - \eta_t^{-1} \nabla h(\pi^t) \right\|^2_{L_2(v_t)} \right] = 2\mathbb{E}_{(s,a)\sim v^t} \left[\phi(s,a)\phi(s,a)^\top \right],$$

and is strongly convex when $\mathbb{E}_{(s,a)\sim v^t}[\phi(s,a)\phi(s,a)^\top]$ is definite positive, which is a standard assumption in the literature on linear function approximation [Agarwal et al., 2020, 2021]. This assumption is easily satisfied. For instance, Cayci et al. [2021, Proposition 3] shows that with v^t chosen as the stationary distribution $d^t_\mu \cdot \pi^t$ over $\mathcal{S} \times \mathcal{A}$ and $\phi(s,a) \sim \mathcal{N}(0, \mathbf{I}_k)$ sampled as Gaussian random features, the assumption holds with high probability guarantee. More generally, with $k \ll |\mathcal{S}||\mathcal{A}|$, it is easy to find k linearly independent $\phi(s,a)$ among all $|\mathcal{S}||\mathcal{A}|$ features such that $\mathbb{E}_{(s,a)\sim v^t} \left[\phi(s,a)\phi(s,a)^\top \right]$ has full rank.

On the contrary, one can compute the convexity condition for the optimization problem in (11) when the policy is parameterized as log-linear and verify that it does not hold everywhere. We refer to the attached Julia file for an example.

A.5 Future work

The results we have obtained open up several experimental questions related to the parameterization class and the choice of mirror map in APMO. We leave such questions as an important direction to further support our theoretical findings.

On the other hand, our work also opens several interesting research directions in both algorithmic and theoretical aspects.

From an algorithmic point of view, the updates in Line 1 and 2 of AMPO are not explicit. This might be an issue in practice, especially for large scale RL problems. It would be interesting to design efficient regression solver for minimizing the approximation error in Line 1 of Algorithm 1. For instance, by using the dual averaging algorithm [Beck, 2017, Chapter 4], it could be possible to replace the term $\nabla h(\pi^t_s)$ with f^t_s for all $s \in \mathcal{S}$, to make the computation of the algorithm more efficient. That is, it could be interesting to consider the following variation of Line 1 in Algortihm 1:

$$\left\| f^{t+1} - Q^t - \frac{\eta_{t-1}}{\eta_t} f^t \right\|^2_{L_2(v_t)} \leq \varepsilon_{\text{approx}}. \tag{25}$$

Notice that (25) has the same update as (15), however (25) is not restricted to using the negative entropy mirror map. To efficiently solve the regression problem in Line 1 of Algorithm 1, one may want to apply the modern variance reduction techniques [Nguyen et al., 2017, Cutkosky and Orabona, 2019, Li et al., 2021]. This has been done by Liu et al. [2020] for NPG method.

From a theoretical point of view, it would be interesting to derive a sample complexity analysis for AMPO in specific settings, by leveraging its linear convergence. As mentioned for the linear MDP [Jin et al., 2020] in Appendix A.2, one can apply the linear convergence theory of AMPO to other structural MDPs, e.g., block MDP [Du et al., 2019a], factored MDP [Kearns and Koller, 1999, Sun et al., 2019], RKHS linear MDP and RKHS linear mixture MDP [Du et al., 2021], to build new sample complexity results for these settings, since the assumptions of Theorem 4.3 do not impose any constraint on the MDP. On the other hand, it would be interesting to explore the interaction between the Bregman projected policy class and the expected Lipschitz and smooth policies [Yuan et al., 2022] and the Fish-non-degenerate policies [Liu et al., 2020] to establish new improved sample complexity results in these settings, again thanks to the linear convergence theory of AMPO.

Additionally, it would be interesting to study the application of AMPO to the offline setting. In the main text, we have discussed how to extend Algorithm 1 and Theorem 4.3 to the offline

setting, where v^t can be set as the state-action distribution induced by an arbitrary behavior policy that generates the data. However, we believe that this direction requires further investigation. One of the major challenges of offline RL is dealing with the distribution shifts that stem from the mismatch between the trained policy π^t and the behaviour policy. Several methods have been introduced to deal with this issue, such as constraining the current policy to be close to the behavior policy [Levine et al., 2020]. We leave introducing offline RL techniques in AMPO as future work.

Another direction for future work is extending the policy update of AMPO to mirror descent algorithm based on value iteration and Bellman operators, such as mirror descent modified policy iteration [Geist et al., 2019], in order to extend existing results to the general parametrization setting.

Finally, this work theoretically indicates that, perhaps the most important future work of PMD-type algorithms is to design efficient policy evaluation algorithms to make the estimation of the Q-function as accurate as possible, such as using offline data for training, and to construct adaptive representation learning for $\mathcal{F}^\Theta$ to closely approximate Q-function, so that ϵ_{approx} is guaranteed to be small. This matches one of the most important research questions for deep Q-learning type algorithms for general policy optimization problems.

B Equivalence of (9)-(10) and (11) in the tabular case

To demonstrate the equivalence between the two-step update (9)-(10) and the one-step update (11) for policy mirror descent in the tabular case, it is sufficient to validate the following lemma, which comes from the optimization literature. The proof of this lemma can be found in Bubeck [2015, Chapter 4.2]. However, for the sake of completeness, we present the proof here.

Lemma B.1 (Right after Theorem 4.2 in Bubeck [2015])**.** *Consider the mirror descent update in (5)-(6) for the minimization of a function $V(\cdot)$, that is,*

$$y^{t+1} = \nabla h(x^t) - \eta_t \nabla V(x)|_{x=x^t}, \tag{26}$$

$$x^{t+1} = Proj^h_{\mathcal{X}}(\nabla h^*(y^{t+1})). \tag{27}$$

Then the mirror descent update can be rewritten as

$$x^{t+1} = \operatorname*{argmin}_{x\in\mathcal{X}} \eta_t \langle x, \nabla V(x)|_{x=x^t}\rangle + \mathcal{D}_h(x, x^t). \tag{28}$$

Proof. From definition of the Bregman projection step, starting from (26) we have

$$\begin{aligned}
x^{t+1} &= \text{Proj}^h_{\mathcal{X}}(\nabla h^*(y^{t+1})) = \operatorname*{argmin}_{x\in\mathcal{X}} \mathcal{D}_h(x, \nabla h^*(y^{t+1})) \\
&= \operatorname*{argmin}_{x\in\mathcal{X}} \nabla h(x) - \nabla h(\nabla h^*(y^{t+1})) - \left\langle \nabla h(\nabla h^*(y^{t+1})), x - \nabla h^*(y^{t+1})\right\rangle \\
&\overset{(4)}{=} \operatorname*{argmin}_{x\in\mathcal{X}} \nabla h(x) - y^{t+1} - \left\langle y^{t+1}, x - \nabla h^*(y^{t+1})\right\rangle \\
&= \operatorname*{argmin}_{x\in\mathcal{X}} \nabla h(x) - \langle x, y^{t+1}\rangle \\
&\overset{(26)}{=} \operatorname*{argmin}_{x\in\mathcal{X}} \nabla h(x) - \langle x, \nabla h(x^t) - \eta_t \nabla V(x)|_{x=x^t}\rangle \\
&= \operatorname*{argmin}_{x\in\mathcal{X}} \eta_t \langle x, \nabla V(x)|_{x=x^t}\rangle + \nabla h(x) - \nabla h(x^t) - \left\langle \nabla h(x^t), x - x^t\right\rangle \\
&= \operatorname*{argmin}_{x\in\mathcal{X}} \eta_t \langle x, \nabla V(x)|_{x=x^t}\rangle + \mathcal{D}_h(x, x^t),
\end{aligned}$$

where the second and the last lines are both obtained by the definition of the Bregman divergence. □

The one-step update in (28) is often taken as the definition of mirror descent [Beck and Teboulle, 2003], which provides a proximal view point of mirror descent, i.e., a gradient step in the primal space with a regularization of Bregman divergence.

C AMPO for specific mirror maps

In this section, we give the derivations for Example 3.2, which is based on the Karush-Kuhn-Tucker (KKT) conditions [Karush, 1939, Kuhn and Tucker, 1951], and then provide details about the ω-potential mirror map class from Section 3.1.

C.1 Derivation of Example 3.2

We give here the derivation of Example 3.2. Let h be the negative entropy mirror map, that is

$$h(\pi_s) = \sum_{a \in \mathcal{A}} \pi(a|s) \log(\pi(a|s)).$$

We solve the minimization problem

$$\pi_s^\theta = \operatorname*{argmin}_{\pi_s \in \Delta(\mathcal{A})} \mathcal{D}_h(\pi_s, \nabla h^*(\eta_{t-1} f_s^t))$$

through the KKT conditions. We formalize it as

$$\begin{aligned} &\operatorname*{argmin}_{\pi_s \in \mathbb{R}^{|\mathcal{A}|}} \ \mathcal{D}_h(\pi_s, \nabla h^*(\eta_{t-1} f_s^t)) \\ &\text{subject to } \langle \pi_s, \mathbf{1} \rangle = 1 \\ &\qquad\qquad \pi(a|s) \geq 0 \qquad \forall\ a \in \mathcal{A}, \end{aligned}$$

where $\mathbf{1}$ denotes a vector in $\mathbb{R}^{|\mathcal{A}|}$ with coordinates equal to 1 element-wisely. The conditions then become

$$\begin{aligned} \text{(stationarity)} \qquad & \log(\pi_s) - \eta_{t-1} f_s^t + \lambda_s \mathbf{1} - \sum_{a \in \mathcal{A}} c_s^a e_a = 0, \\ \text{(complementary slackness)} \qquad & c_s^a \pi(a|s) = 0 \quad \forall\ a \in \mathcal{A}, \\ \text{(primal feasibility)} \qquad & \langle \pi_s, \mathbf{1} \rangle = 1,\ \pi(a|s) \geq 0 \qquad \forall\ a \in \mathcal{A}, \\ \text{(dual feasibility)} \qquad & c_s^a \geq 0 \quad \forall\ a \in \mathcal{A}, \end{aligned}$$

where $\log(\pi_s)$ is applied element-wisely, λ_s and $(c_s^a)_{a \in \mathcal{A}}$ are the dual variables, and e_a is a vector with zero entries except one non-zero entry 1 corresponding to the action a, for all $a \in \mathcal{A}$. It is easy to verify that the solution

$$\pi_s^\theta = \frac{\exp(\eta f_s^\theta)}{\|\exp(\eta f_s^\theta)\|_1},$$

with $\lambda_s = \log \sum_{a \in \mathcal{A}} \exp_q(\eta_{t-1} f^t(s,a))$ and $c_s^a = 0$ for all $a \in \mathcal{A}$, satisfies all the conditions.

When $f^\theta(s,a) = \theta_{s,a}$ we obtain the tabular softmax policy $\pi^\theta(a|s) \propto \exp(\eta \theta_{s,a})$. When $f^\theta(s,a) = \theta^\top \phi(s,a)$ is a linear function, for $\theta \in \mathbb{R}^d$ and for a feature function $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$, we obtain the log-linear policy $\pi^\theta(a|s) \propto \exp(\eta \theta^\top \phi(s,a))$. When $f^\theta : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a neural network, we obtain the softmax neural policy $\pi^\theta(a|s) \propto \exp(\eta f(s,a))$.

C.2 More on ω-potential mirror maps

In this section, we provide details about the ω-potential mirror map class from Section 3.1, including the derivation of (12), several instantiations of ω-potential mirror map mentioned in Section 3.1 with their derivations, and an iterative algorithm to find approximately the Bregman projection induced by ω-potential mirror map when an exact solution is not available.

We give a different but equivalent formulation of Proposition 2 of Krichene et al. [2015].

Proposition C.1. *For $a \in (-\infty, +\infty]$ and $\omega \leq 0$, an increasing C^1-diffeomorphism $\phi : (-\infty, a) \to (\omega, +\infty)$ is called an ω-potential if*

$$\lim_{u \to -\infty} \phi(u) = \omega, \quad \lim_{u \to a} \phi(u) = +\infty, \quad \int_0^1 \phi^{-1}(u) du \leq \infty.$$

Let the mirror map h_ϕ be defined as

$$h_\phi(\pi_s) = \sum_{a\in\mathcal{A}} \int_1^{\pi(a|s)} \phi^{-1}(u)du.$$

We have then that π_s^t is a solution to the Bregman projection

$$\min_{\pi\in\Delta_s} Proj^h_{\Delta(\mathcal{A})}(\nabla h^*(\eta_{t-1}f_s^t)),$$

if and only if there exist a normalization constant $\lambda_s \in \mathbb{R}$ such that

$$\pi^t(a|s) = \sigma(\phi(\eta_{t-1}f^t(s,a) + \lambda_s)) \quad a \in \mathcal{A},$$

and $\sum_{a\in\mathcal{A}} \pi^t(a|s) = 1$, where for all $s \in \mathcal{S}$ and $\sigma(z) = \max(z, 0)$ for $z \in \mathbb{R}$.

We can now use Proposition C.1 to instantiate AMPO for mirror maps belonging to this class. We highlight that due to the definition of the Bregman divergence, two mirror maps that only differ for a constant term are equivalent and generate the same algorithm. We start with the negative entropy, which leads to a closed solution for λ_s and therefore for the Bregman projection.

Negative entropy Let $\phi(u) = \exp(u-1)$. The mirror map h_ϕ becomes the negative entropy, as

$$h_\phi(\pi_s) = \sum_{a\in\mathcal{A}} \int_1^{\pi(a|s)} (\log(u)+1)du = \sum_{a\in\mathcal{A}} \pi(a|s)\log(\pi(a|s)),$$

and the associated Bregman divergence becomes the KL divergence, i.e., $D_{h_\phi}(\pi_s, \bar{\pi}_s) = \mathtt{KL}(\pi_s, \bar{\pi}_s)$. Equation (15) follows from Equation (12) and the fact that $\phi^{-1}(0) = -\infty$, which means that $\max(\eta_{t-1}f^t, \phi^{-1}(0) - \lambda_s^t) = \eta_{t-1}f^t$. As we showed in Appendix C.1, the Bregman projection for the negative entropy has a closed form.

We next present the squared ℓ_2-norm and the ϵ-negative entropy. For these two mirror maps, the Bregman projection can be computed exactly but has no closed form.

Squared ℓ_2-norm Let ϕ be the identity function. The mirror map h_ϕ becomes the squared ℓ_2-norm, up to a constant term, as

$$h_\phi(\pi_s) = \sum_{a\in\mathcal{A}} \int_1^{\pi(a|s)} u\ du = \frac{1}{2}\left(\sum_{a\in\mathcal{A}} \pi(a|s)^2 - 1\right),$$

the associated Bregman divergence becomes the Euclidean distance, i.e., $D_{h_\phi}(\pi_s, \bar{\pi}_s) = \|\pi_s - \bar{\pi}_s\|_2^2$, and $\nabla h^*(\cdot)$ is the identity function. The update in (13) follows immediately and the projection step with the Euclidean distance becomes, for all $s \in \mathcal{S}$,

$$\pi_s^{t+1} = \mathrm{Proj}^{l_2}_{\Delta(\mathcal{A})}(\nabla h^*(\eta_t f_s^{t+1})) = \mathrm{Proj}^{l_2}_{\Delta(\mathcal{A})}(\eta_t f_s^{t+1}) = \operatorname*{argmin}_{\pi\in\Delta(\mathcal{A})} \left\|\pi - \eta_t f_s^{t+1}\right\|_2^2, \tag{29}$$

giving in Equations (13)-(14) a generalization of the projected-Q descent algorithm developed by Xiao [2022] for tabular policies. The Euclidean projection onto the probability simplex can be obtained exactly, as shown by Wang and Carreira-Perpinán [2013].

ϵ-negative entropy [Krichene et al., 2015] Let $\epsilon \geq 0$ and define the ϵ-exponential potential as $\phi(x) = \exp(x-1) - \epsilon$. The mirror map h_ϕ becomes

$$h_\phi(\pi_s) = \sum_{a\in\mathcal{A}} \int_1^{\pi(a|s)} (\log(u+\epsilon)+1)du = \sum_{a\in\mathcal{A}} \left[(\pi(a|s)+\epsilon)\ln(\pi(a|s)+\epsilon) - (1+\epsilon)\ln(1+\epsilon)\right].$$

An exact solution to the associated projection can then be found in $\widetilde{\mathcal{O}}(|\mathcal{A}|)$ computations using Algorithm 2, which has been proposed by Krichene et al. [2015, Algorithm 4]. Additionally, following (12), the regression problem in Line 1 of Algorithm 1 becomes

$$\theta^{t+1} \in \operatorname*{argmin}_{\theta\in\Theta} \left\|f^\theta - Q^t - \eta_t^{-1}\max(\eta_{t-1}f^t, 1 + \log(\epsilon) - \lambda_s^t)\right\|^2_{L_2(v_t)},$$

Algorithm 2: Bregman projection for ϵ-negative entropy

Input: vector to project $x \in \mathbb{R}^{|\mathcal{A}|}$, parameter ϵ.

Initialize $y = \exp(x)$ element-wisely.

Let $y^{(i)}$ be the i-th smallest element of y.

Let $i^\star$ be the smallest index for which

$$(1 + \epsilon(|\mathcal{A}| - i + 1))y^{(i)} - \epsilon \sum_{j \geq i} y^{(j)} > 0.$$

Set

$$\lambda = \frac{\sum_{i \geq i^\star} y^{(i)}}{1 + \epsilon(|\mathcal{A}| - i^\star + 1)}.$$

Return: the projected vector $\left(\sigma\left(-\epsilon + y_a/\lambda\right)\right)_{a \in \mathcal{A}}$.

Algorithm 3: Bregman projection for ω-potential mirror maps

Input: vector to project $x \in \mathbb{R}^{|\mathcal{A}|}$, ω-potential ϕ, precision ε.

Initialize

$$\bar{\nu} = \phi^{-1}(1) - \max_{a \in \mathcal{A}} x_a$$
$$\underline{\nu} = \phi^{-1}(1/|\mathcal{A}|) - \max_{a \in \mathcal{A}} x_a$$

Define $\tilde{x}(\nu) = \left(\sigma(\phi\left(x_a + \nu\right))\right)_{a \in \mathcal{A}}$.

while $\|\tilde{x}(\bar{\nu}) - \tilde{x}(\underline{\nu})\|_1 > \varepsilon$ **do**

  Let $\nu^+ \leftarrow (\bar{\nu} + \underline{\nu})/2$

  if $\sum_{a \in \mathcal{A}} \tilde{x}_a(\nu^+) > 1$ **then**

    $\bar{\nu} \leftarrow \nu^+$

  else

    $\underline{\nu} \leftarrow \nu^+$

Return $\tilde{x}(\bar{\nu})$

where λ_s^t can be obtained through Algorithm 2.

The Bregman projection for generic mirror maps can be computed approximately in $\widetilde{\mathcal{O}}(|\mathcal{A}|)$ computations through a bisection algorithm. Krichene et al. [2015] propose one such algorithm, which we report in Algorithm 3 for completeness. We next provide two mirror maps that have appeared before in the optimization literature, but do not lead to an exact solution to the Bregman projection. We leave them as object for future work.

Negative Tsallis entropy [Orabona, 2020] Let $q > 0$ and define ϕ as

$$\phi_q(u) = \begin{cases} \exp(u-1) & \text{if } q = 1, \\ \left[\sigma\left(\frac{(q-1)u}{q}\right)\right]^{\frac{1}{q-1}} & \text{else.} \end{cases}$$

The mirror map h_{ϕ_q} becomes the negative Tsallis entropy, that is

$$h_{\phi_q}(\pi_s) = \sum \pi(a|s) \log_q(\pi(a|s)),$$

where, for $y > 0$,

$$\log_q(y) = \begin{cases} \log(y) & \text{if } q = 1, \\ \frac{-y^{q-1}}{q-1} & \text{else.} \end{cases}$$

If $q \neq 1$ and following (12), the regression problem in Line 1 of Algorithm 1 becomes

$$\theta^{t+1} \in \operatorname*{argmin}_{\theta \in \Theta} \left\| f^\theta - Q^t - \eta_t^{-1} \max(\eta_{t-1} f^t, -\lambda_s^t) \right\|_{L_2(v_t)}^2,$$

Hyperbolic entropy [Ghai et al., 2020] Let $b > 0$ and define ϕ as

$$\phi_b(u) = b \sinh(u)$$

The mirror map h_{ϕ_b} becomes the hyperbolic entropy, that is

$$h_{\phi_b}(\pi_s) = \sum_{a \in \mathcal{A}} \pi(a|s) \operatorname{arcsinh}(\pi(a|s)/b) - \sqrt{\pi(a|s)^2 + b^2},$$

and, following (12), the regression problem in Line 1 of Algorithm 1 becomes

$$\theta^{t+1} \in \operatorname*{argmin}_{\theta \in \Theta} \left\| f^\theta - Q^t - \eta_t^{-1} \max(\eta_{t-1} f^t, -\lambda_s^t) \right\|_{L_2(v_t)}^2 .$$

Regarding the limitations of the ω-potential mirror map class, we are aware of two previously used mirror maps that cannot be recovered using ω-potentials: $h(x) = \frac{1}{2} x^\top A x$, for some matrix A, which generates the Mahalanobis distance, and p-norms, i.e. $h(x) = \|x\|_p^2$. Note that the case where $h(x) = \|x\|_p^p$ can be recovered.

D Deferred proofs from Section 4.1

D.1 Proof of Lemma 4.1

Here we provide the proof of Lemma 4.1, a variant of the three-point descent lemma with the integration of an arbitrary parameterized function, which is the key tool for our analysis of AMPO. It is a variation of both Xiao [2022, Lemma 6] and Chen and Teboulle [1993, Lemma 3.2]. First, we recall some technical conditions of the mirror map [Bubeck, 2015, Chapter 4].

Suppose that $\mathcal{Y} \subset \mathbb{R}^{|\mathcal{A}|}$ is a closed convex set, we say a function $h : \mathcal{Y} \to \mathbb{R}$ is a *mirror map* if it satisfies the following properties:

(i) h is strictly convex and differentiable;

(ii) h is essentially smooth, i.e., the graident of h diverges on the boundary of $\mathcal{Y}$, that is $\lim_{x \to \partial \mathcal{Y}} \|\nabla h(x)\| \to \infty$;

(iii) the gradient of h takes all possible values, that is $\nabla h(\mathcal{Y}) = \mathbb{R}^{|\mathcal{A}|}$.

To prove Lemma 4.1, we also need the following rather simple properties, i.e., the three-point identity and the generalized Pythagorean theorem, satisfied by the Bregman divergence. We provide their proofs for self-containment.

Lemma D.1 (Three-point identity, Lemma 3.1 in Chen and Teboulle [1993])**.** *Let h be a mirror map. For any a, b in the relative interior of $\mathcal{Y}$ and $c \in \mathcal{Y}$, we have that:*

$$\mathcal{D}_h(c, a) + \mathcal{D}_h(a, b) - \mathcal{D}_h(c, b) = \langle \nabla h(b) - \nabla h(a), c - a \rangle . \tag{30}$$

Proof. Using the definition of the Bregman divergence $\mathcal{D}_h$, we have

$$\langle \nabla h(a), c - a \rangle = h(c) - h(a) - \mathcal{D}_h(c, a), \tag{31}$$
$$\langle \nabla h(b), a - b \rangle = h(a) - h(b) - \mathcal{D}_h(a, b), \tag{32}$$
$$\langle \nabla h(b), c - b \rangle = h(c) - h(b) - \mathcal{D}_h(c, b). \tag{33}$$

Subtracting (31) and (32) from (33) yields (30). □

Lemma D.2 (Generalized Pythagorean Theorem of Bregman divergence, Lemma 4.1 in Bubeck [2015])**.** *Let $\mathcal{X} \subseteq \mathcal{Y}$ be a closed convex set. Let h be a mirror map defined on $\mathcal{Y}$. Let $x \in \mathcal{X}$, $y \in \mathcal{Y}$ and $y^\star = Proj_{\mathcal{X}}^h(y)$, then*

$$\langle \nabla h\left(y^\star\right) - \nabla(y), y^\star - x \rangle \leq 0,$$

which also implies

$$\mathcal{D}_h\left(x, y^\star\right) + \mathcal{D}_h\left(y^\star, y\right) \leq \mathcal{D}_h(x, y). \tag{34}$$

Proof. From the definition of $y^\star$, which is

$$y^\star \in \operatorname*{argmin}_{y' \in \mathcal{X}} \mathcal{D}_h(y', y),$$

and from the first-order optimality condition [Bubeck, 2015, Proposition 1.3], with

$$\nabla_{y'} \mathcal{D}_h(y', y) = \nabla h(y') - \nabla h(y), \quad \text{for all } y' \in \mathcal{Y},$$

we have

$$\langle \nabla_{y'} \mathcal{D}_h(y', y)|_{y'=y^\star}, y^\star - x \rangle \leq 0 \quad \Longrightarrow \quad \langle \nabla h\left(y^\star\right) - \nabla(y), y^\star - x \rangle \leq 0,$$

which implies (34) by applying the definition of Bregman divergence and rearranging terms. □

Now we are ready to prove Lemma 4.1.

Lemma D.3 (Lemma 4.1)**.** *Let $\mathcal{Y} \subset \mathbb{R}^{|\mathcal{A}|}$ be a closed convex set with $\Delta(\mathcal{A}) \subseteq \mathcal{Y}$. For any policies $\pi \in \Delta(\mathcal{A})$ and $\bar{\pi}$ in the relative interior of $\Delta(\mathcal{A})$, any function f^θ with $\theta \in \Theta$, any $s \in \mathcal{S}$ and for $\eta > 0$, we have that,*

$$\langle \eta f_s^\theta - \nabla h(\bar{\pi}_s), \pi_s - \tilde{\pi}_s \rangle \leq \mathcal{D}_h(\pi_s, \bar{\pi}_s) - \mathcal{D}_h(\tilde{\pi}_s, \bar{\pi}_s) - \mathcal{D}_h(\pi, \tilde{\pi}_s),$$

where $\tilde{\pi}$ is induced by f^θ and η according to Definition 3.1, that is, for all $s \in \mathcal{S}$,

$$\tilde{\pi}_s = Proj^h_{\Delta(\mathcal{A})}\left(\nabla h^*(\eta f_s^\theta)\right) = \operatorname*{argmin}_{\pi'_s \in \Delta(\mathcal{A})} \mathcal{D}_h(\pi'_s, \nabla h^*(\eta f_s^\theta)). \tag{35}$$

Proof. For clarity of exposition, let $p_s = \nabla h^*(\eta f_s^\theta)$. Plugging $a = \bar{\pi}_s$, $b = p_s$ and $c = \pi_s$ in the three-point identity lemma D.1, we obtain

$$\mathcal{D}_h(\pi_s, \bar{\pi}_s) - \mathcal{D}_h(\pi_s, p_s) + \mathcal{D}_h(\bar{\pi}_s, p_s) = \langle \nabla h(\bar{\pi}_s) - \nabla h(p_s), \bar{\pi}_s - \pi_s \rangle . \tag{36}$$

Similarly, plugging $a = \bar{\pi}_s, b = p_s$ and $c = \tilde{\pi}_s$ in the three-point identity lemma D.1, we obtain

$$\mathcal{D}_h(\tilde{\pi}_s, \bar{\pi}_s) - \mathcal{D}_h(\tilde{\pi}_s, p_s) + \mathcal{D}_h(\bar{\pi}_s, p_s) = \langle \nabla h(\bar{\pi}_s) - \nabla h(p_s), \bar{\pi}_s - \tilde{\pi}_s \rangle . \tag{37}$$

From (36), we have

$$\begin{aligned}
& \mathcal{D}_h(\pi_s, \bar{\pi}_s) - \mathcal{D}_h(\pi_s, p_s) + \mathcal{D}_h(\bar{\pi}_s, p_s) \\
= \; & \langle \nabla h(\bar{\pi}_s) - \nabla h(p_s), \bar{\pi}_s - \pi_s \rangle \\
= \; & \langle \nabla h(\bar{\pi}_s) - \nabla h(p_s), \bar{\pi}_s - \tilde{\pi}_s \rangle + \langle \nabla h(\bar{\pi}_s) - \nabla h(p_s), \tilde{\pi}_s - \pi_s \rangle \\
\overset{(37)}{=} \; & \mathcal{D}_h(\tilde{\pi}_s, \bar{\pi}_s) - \mathcal{D}_h(\tilde{\pi}_s, p_s) + \mathcal{D}_h(\bar{\pi}_s, p_s) + \langle \nabla h(\bar{\pi}_s) - \nabla h(p_s), \tilde{\pi}_s - \pi_s \rangle .
\end{aligned}$$

By rearranging terms, we have

$$\mathcal{D}_h(\pi_s, \bar{\pi}_s) - \mathcal{D}_h(\tilde{\pi}_s, \bar{\pi}_s) - \mathcal{D}_h(\pi_s, p_s) + \mathcal{D}_h(\tilde{\pi}_s, p_s) = \langle \nabla h(\bar{\pi}_s) - \nabla h(p_s), \tilde{\pi}_s - \pi_s \rangle . \tag{38}$$

From the Generalized Pythagorean Theorem of the Bregman divergence in Lemma D.2, also known as non-expansivity property, and from the fact that $\tilde{\pi}_s = \text{Proj}^h_{\Delta(\mathcal{A})}(p_s)$, we have that

$$\mathcal{D}_h(\pi_s, \tilde{\pi}_s) + \mathcal{D}_h(\tilde{\pi}_s, p_s) \leq \mathcal{D}_h(\pi_s, p_s) \quad \Longleftrightarrow \quad -\mathcal{D}_h(\pi_s, p_s) + \mathcal{D}_h(\tilde{\pi}_s, p_s) \leq -\mathcal{D}_h(\pi_s, \tilde{\pi}_s).$$

Plugging the above inequality into the left hand side of (38) yields

$$\mathcal{D}_h(\pi_s, \bar{\pi}_s) - \mathcal{D}_h(\tilde{\pi}_s, \bar{\pi}_s) - \mathcal{D}_h(\pi_s, \tilde{\pi}_s) \geq \langle \nabla h(\bar{\pi}_s) - \nabla h(p_s), \tilde{\pi}_s - \pi_s \rangle ,$$

which concludes the proof with $\nabla h(p_s) = \eta f_s^\theta$. □

We also provide a sketch of an alternative proof for Lemma 4.1, which involves the three-point descent lemma from Xiao [2022]. Starting from the definition of $\tilde{\pi}$, we have

$$\begin{aligned}
\tilde{\pi}_s &= \operatorname*{argmin}_{\pi' \in \Delta(\mathcal{A})} \mathcal{D}_h(\pi', \nabla h^*(\eta f_s^\theta)) \\
&= \operatorname*{argmin}_{\pi' \in \Delta(\mathcal{A})} h(\pi') - h(\nabla h^*(\eta f_s^\theta)) - \langle \nabla h(\nabla h^*(\eta f_s^\theta)), \pi' - \nabla h^*(\eta f_s^\theta) \rangle \\
&= \operatorname*{argmin}_{\pi' \in \Delta(\mathcal{A})} h(\pi') - \langle \eta f_s^\theta, \pi' \rangle \\
&= \operatorname*{argmin}_{\pi' \in \Delta(\mathcal{A})} \langle -\eta f_s^\theta + \nabla h(\bar{\pi}_s), \pi' \rangle + h(\pi') - h(\bar{\pi}_s) - \langle \nabla h(\bar{\pi}_s), \pi' \rangle \\
&= \operatorname*{argmin}_{\pi' \in \Delta(\mathcal{A})} \langle -\eta f_s^\theta + \nabla h(\bar{\pi}_s), \pi' \rangle + \mathcal{D}_h(\bar{\pi}_s, \pi'),
\end{aligned} \tag{39}$$

where the second and the last lines are obtained using the definition of the Bregman divergence, and the third line is obtained using (4) ($\nabla h(\nabla h^*(x^*)) = x^*$ for all $x^* \in \mathbb{R}^{|\mathcal{A}|}$). Lemma 4.1 is obtained by applying the three-point descent lemma in Xiao [2022] to (39).

D.2 Bounding errors

In this section, we will bound error terms of the type

$$\mathbb{E}_{s \sim d_\mu^\pi, a \sim \pi_s} \left[Q^t(s,a) + \eta_t^{-1} [\nabla h(\pi_s^t)]_a - f^{t+1}(s,a) \right], \tag{40}$$

where $(d_\mu^\pi, \pi) \in \{(d_\mu^\star, \pi^\star), (d_\mu^{t+1}, \pi^{t+1}), (d_\mu^\star, \pi^t), (d_\mu^{t+1}, \pi^t)\}$. These error terms appear in the forthcoming proofs of our results and directly induce the error floors in the convergence rates.

In the rest of Appendix D, let $q^t : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ such that, for every $s \in \mathcal{S}$,

$$q_s^t := f_s^{t+1} - \eta_t^{-1} \nabla h(\pi_s^t) \in \mathbb{R}^{|\mathcal{A}|}.$$

So (40) can be rewritten as

$$\mathbb{E}_{s \sim d_\mu^\pi, a \sim \pi_s} \left[Q^t(s,a) + \eta_t^{-1} [\nabla h(\pi_s^t)]_a - f^{t+1}(s,a) \right] = \mathbb{E}_{s \sim d_\mu^\pi, a \sim \pi_s} \left[Q^t(s,a) - q^t(s,a) \right]. \tag{41}$$

To bound it, let $(v^t)_{t \geq 0}$ be a sequence of distributions over states and actions. By using Cauchy-Schwartz's inequality, we have

$$\begin{aligned}
&\mathbb{E}_{s \sim d_\mu^\pi, a \sim \pi_s} \left[Q^t(s,a) - q^t(s,a) \right] \\
&\quad = \int_{s \in \mathcal{S}, a \in \mathcal{A}} \frac{d_\mu^\pi(s) \pi(a \mid s)}{\sqrt{v^t(s,a)}} \cdot \sqrt{v^t(s,a)} (Q^t(s,a) - q^t(s,a)) \\
&\quad \leq \sqrt{\int_{s \in \mathcal{S}, a \in \mathcal{A}} \frac{\left(d_\mu^\pi(s) \pi(a \mid s)\right)^2}{v^t(s,a)} \cdot \int_{s \in \mathcal{S}, a \in \mathcal{A}} v^t(s,a) (Q^t(s,a) - q^t(s,a))^2} \\
&\quad = \sqrt{\mathbb{E}_{(s,a) \sim v^t} \left[\left(\frac{d_\mu^\pi(s) \pi(a \mid s)}{v^t(s,a)} \right)^2 \right] \cdot \mathbb{E}_{(s,a) \sim v^t} \left[(Q^t(s,a) - q^t(s,a))^2 \right]} \\
&\quad \leq \sqrt{C_v \mathbb{E}_{(s,a) \sim v^t} \left[(Q^t(s,a) - q^t(s,a))^2 \right]},
\end{aligned}$$

where the last line is obtained by Assumption (A2). Using the concavity of the square root and Assumption (A1), we have that

$$\mathbb{E} \left[\mathbb{E}_{s \sim d_\mu^\pi, a \sim \pi_s} \left[Q^t(s,a) - q^t(s,a) \right] \right] \leq \sqrt{C_v \varepsilon_{\text{approx}}}. \tag{42}$$

D.3 Quasi-monotonic updates – Proof of Proposition 4.2

In this section, we show that the AMPO updates guarantee a quasi-monotonic property, i.e., a non-decreasing property up to a certain error floor due to the approximation error, which

allows us to establish an important recursion about the AMPO iterates next. First, we recall the performance difference lemma [Kakade and Langford, 2002] which is the second key tool for our analysis and a well known result in the RL literature. Here we use a particular form of the lemma presented by Xiao [2022, Lemma 1].

Lemma D.4 (Performance difference lemma, Lemma 1 in [Xiao, 2022])**.** *For any policy* $\pi, \pi' \in \Delta(\mathcal{A})^{\mathcal{S}}$ *and* $\mu \in \Delta(\mathcal{S})$*,*

$$V^{\pi}(\mu) - V^{\pi'}(\mu) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_{\mu}^{\pi}}\left[\left\langle Q_s^{\pi'}, \pi_s - \pi'_s\right\rangle\right].$$

For clarity of exposition, we introduce the notation

$$\tau := \frac{2\sqrt{C_v \varepsilon_{\text{approx}}}}{1-\gamma}.$$

The following result characterizes the non-decreasing property of AMPO. The error bound (42) in the Appendix D.2 will be used to prove the lemma. It is a slightly stronger result than Proposition 4.2.

Lemma D.5. *For the iterates of Algorithm 1, at each time* $t \geq 0$*, we have*

$$\mathbb{E}[V^{t+1}(\mu) - V^{t}(\mu)] \geq \mathbb{E}\left[\mathbb{E}_{s\sim d_{\mu}^{t+1}}\left[\frac{\mathcal{D}_h(\pi_s^{t+1}, \pi_s^t) + \mathcal{D}_h(\pi_s^t, \pi_s^{t+1})}{\eta_t(1-\gamma)}\right]\right] - \tau.$$

Proof. Using Lemma 4.1 with $\bar{\pi} = \pi^t$, $f^{\theta} = f^{t+1}$, $\eta = \eta_t$, thus $\tilde{\pi} = \pi^{t+1}$ by Definition 3.1 and Algorithm 1, and $\pi_s = \pi_s^t$, we have

$$\langle \eta_t q_s^t, \pi_s^t - \pi_s^{t+1}\rangle \leq \mathcal{D}_h(\pi_s^t, \pi_s^t) - \mathcal{D}_h(\pi_s^{t+1}, \pi_s^t) - \mathcal{D}_h(\pi_s^t, \pi_s^{t+1}). \tag{43}$$

By rearranging terms and noticing $\mathcal{D}_h(\pi_s^t, \pi_s^t) = 0$, we have

$$\langle \eta_t q_s^t, \pi_s^{t+1} - \pi_s^t\rangle \geq \mathcal{D}_h(\pi_s^{t+1}, \pi_s^t) + \mathcal{D}_h(\pi_s^t, \pi_s^{t+1}) \geq 0. \tag{44}$$

Then, by the performance difference lemma D.4, we have

$$\begin{aligned}
(1-\gamma)\mathbb{E}[V^{t+1}(\mu) - V^t(\mu)] &= \mathbb{E}\left[\mathbb{E}_{s\sim d_{\mu}^{t+1}}\left[\langle Q_s^t, \pi_s^{t+1} - \pi_s^t\rangle\right]\right] \\
&= \mathbb{E}\left[\mathbb{E}_{s\sim d_{\mu}^{t+1}}\left[\langle q_s^t, \pi_s^{t+1} - \pi_s^t\rangle\right]\right] \\
&\quad + \mathbb{E}\left[\mathbb{E}_{s\sim d_{\mu}^{t+1}}\left[\langle Q_s^t - q_s^t, \pi_s^{t+1} - \pi_s^t\rangle\right]\right] \\
&\overset{(43)}{\geq} \mathbb{E}\left[\mathbb{E}_{s\sim d_{\mu}^{t+1}}\left[\frac{\mathcal{D}_h(\pi_s^{t+1}, \pi_s^t) + \mathcal{D}_h(\pi_s^t, \pi_s^{t+1})}{\eta_t}\right]\right] \\
&\quad - \left|\mathbb{E}\left[\mathbb{E}_{s\sim d_{\mu}^{t+1}}\left[\langle Q_s^t - q_s^t, \pi_s^{t+1} - \pi_s^t\rangle\right]\right]\right| \\
&\geq \mathbb{E}\left[\mathbb{E}_{s\sim d_{\mu}^{t+1}}\left[\frac{\mathcal{D}_h(\pi_s^{t+1}, \pi_s^t) + \mathcal{D}_h(\pi_s^t, \pi_s^{t+1})}{\eta_t}\right]\right] - \tau(1-\gamma),
\end{aligned}$$

which concludes the proof after dividing both sides by $(1-\gamma)$. The last line follows from

$$\begin{aligned}
\left|\mathbb{E}\left[\mathbb{E}_{s\sim d_{\mu}^{t+1}}\left[\langle Q_s^t - q_s^t, \pi_s^{t+1} - \pi_s^t\rangle\right]\right]\right| &\leq \left|\mathbb{E}\left[\mathbb{E}_{s\sim d_{\mu}^{t+1}, a\sim\pi_s^{t+1}}\left[Q^t(s,a) - q^t(s,a)\right]\right]\right| \quad (45) \\
&\quad + \left|\mathbb{E}\left[\mathbb{E}_{s\sim d_{\mu}^{t+1}, a\sim\pi_s^{t}}\left[Q^t(s,a) - q^t(s,a)\right]\right]\right| \\
&\overset{(42)}{\leq} 2\sqrt{C_1 \varepsilon_{\text{error}}} = \tau(1-\gamma), \quad (46)
\end{aligned}$$

where both terms are upper bounded by $\sqrt{C_v \varepsilon_{\text{approx}}}$ through (42) with $(d_{\mu}^{\pi}, \pi) = (d_{\mu}^{t+1}, \pi^{t+1})$ and $(d_{\mu}^{\pi}, \pi) = (d_{\mu}^{t+1}, \pi^t)$, respectively. □

D.4 Main passage – An important recursion about the AMPO method

In this section, we show an important recursion result for the AMPO updates, which will be used for both the sublinear and the linear convergence analysis of AMPO.

For clarity of exposition in the rest of Appendix D, let

$$\nu_t := \left\| \frac{d_\mu^\star}{d_\mu^{t+1}} \right\|_\infty := \max_{s\in\mathcal{S}} \frac{d_\mu^\star(s)}{d_\mu^{t+1}(s)}.$$

For two different time $t, t' \geq 0$, let $\mathcal{D}_{t'}^t$ denote the expected Bregman divergence between the policy π^t and policy $\pi^{t'}$, where the expectation is taken over the discounted state visitation distribution of the optimal policy $d_\mu^\star$, that is,

$$\mathcal{D}_{t'}^t := \mathbb{E}_{s\sim d_\mu^\star}\left[\mathcal{D}_h(\pi_s^t, \pi_s^{t'})\right].$$

Similarly, let $\mathcal{D}_t^\star$ denote the expected Bregman divergence between the optimal policy $\pi^\star$ and π^t, that is,

$$\mathcal{D}_t^\star := \mathbb{E}_{s\sim d_\mu^\star}\left[\mathcal{D}_h(\pi_s^\star, \pi_s^t)\right].$$

Let $\Delta_t := V^\star(\mu) - V^t(\mu)$ be the optimality gap.

We can now state the following important recursion result for the AMPO method.

Proposition D.6. *Consider the iterates of Algorithm 1, at each time $t \geq 0$, we have*

$$\mathbb{E}\left[\frac{\mathcal{D}_t^{t+1}}{(1-\gamma)\eta_t} + \nu_\mu\left(\Delta_{t+1} - \Delta_t\right) + \Delta_t\right] \leq \mathbb{E}\left[\frac{\mathcal{D}_t^\star}{(1-\gamma)\eta_t} - \frac{\mathcal{D}_{t+1}^\star}{(1-\gamma)\eta_t}\right] + (1+\nu_\mu)\tau.$$

Proof. Using Lemma 4.1 with $\bar{\pi} = \pi^t$, $f^\theta = f^{t+1}$, $\eta = \eta_t$, and thus $\tilde{\pi} = \pi^{t+1}$ by Definition 3.1 and Algorithm 1, and $\pi_s = \pi_s^\star$, we have that

$$\langle \eta_t q_s^t, \pi_s^\star - \pi_s^{t+1}\rangle \leq \mathcal{D}_h(\pi^\star, \pi^t) - \mathcal{D}_h(\pi^\star, \pi^{t+1}) - \mathcal{D}_h(\pi^{t+1}, \pi^t),$$

which can be decomposed as

$$\langle \eta_t q_s^t, \pi_s^t - \pi_s^{t+1}\rangle + \langle \eta_t q_s^t, \pi_s^\star - \pi_s^t\rangle \leq \mathcal{D}_h(\pi^\star, \pi^t) - \mathcal{D}_h(\pi^\star, \pi^{t+1}) - \mathcal{D}_h(\pi^{t+1}, \pi^t).$$

Taking expectation with respect to the distribution $d_\mu^\star$ over states and with respect to the randomness of AMPO and dividing both sides by η_t, we have

$$\mathbb{E}\left[\mathbb{E}_{s\sim d_\mu^\star}\left[\langle q_s^t, \pi_s^t - \pi_s^{t+1}\rangle\right]\right] + \mathbb{E}\left[\mathbb{E}_{s\sim d_\mu^\star}\left[\langle q_s^t, \pi_s^\star - \pi_s^t\rangle\right]\right] \leq \frac{1}{\eta_t}\mathbb{E}[\mathcal{D}_t^\star - \mathcal{D}_{t+1}^\star - \mathcal{D}_t^{t+1}]. \quad (47)$$

We lower bound the two terms on the left hand side of (47) separately. For the first term, we have that

$$\begin{aligned}
\mathbb{E}\left[\mathbb{E}_{s\sim d_\mu^\star}\left[\langle q_s^t, \pi_s^t - \pi_s^{t+1}\rangle\right]\right] &\overset{(44)}{\geq} \left\|\frac{d_\mu^\star}{d_\mu^{t+1}}\right\|_\infty \mathbb{E}\left[\mathbb{E}_{s\sim d_\mu^{t+1}}\left[\langle q_s^t, \pi_s^t - \pi_s^{t+1}\rangle\right]\right] \\
&= \nu_{t+1}\mathbb{E}\left[\mathbb{E}_{s\sim d_\mu^{t+1}}\left[\langle Q_s^t, \pi_s^t - \pi_s^{t+1}\rangle\right]\right] \\
&\quad + \nu_{t+1}\mathbb{E}\left[\mathbb{E}_{s\sim d_\mu^{t+1}}\left[\langle q_s^t - Q_s^t, \pi_s^t - \pi_s^{t+1}\rangle\right]\right] \\
&\overset{(a)}{=} \nu_{t+1}(1-\gamma)\mathbb{E}\left[V^t(\mu) - V^{t+1}(\mu)\right] \\
&\quad + \nu_{t+1}\mathbb{E}\left[\mathbb{E}_{s\sim d_\mu^{t+1}}\left[\langle q_s^t - Q_s^t, \pi_s^t - \pi_s^{t+1}\rangle\right]\right] \\
&\overset{(45)}{\geq} \nu_{t+1}(1-\gamma)\mathbb{E}\left[V^t(\mu) - V^{t+1}(\mu)\right] - \nu_{t+1}\tau(1-\gamma) \\
&= \nu_{t+1}(1-\gamma)\mathbb{E}\left[\Delta_{t+1} - \Delta_t\right] - \nu_{t+1}\tau(1-\gamma),
\end{aligned}$$

where (a) follows from Lemma D.4. For the second term, we have that

$$\begin{aligned}\mathbb{E}\left[\mathbb{E}_{s\sim d_\mu^\star}\left[\langle q_s^t, \pi_s^\star - \pi_s^t\rangle\right]\right] &= \mathbb{E}\left[\mathbb{E}_{s\sim d_\mu^\star}\left[\langle Q_s^t, \pi_s^\star - \pi_s^t\rangle\right]\right] + \mathbb{E}\left[\mathbb{E}_{s\sim d_\mu^\star}\left[\langle q_s^t - Q_s^t, \pi_s^\star - \pi_s^t\rangle\right]\right] \\ &\overset{(b)}{=} \mathbb{E}[\Delta_t](1-\gamma) + \mathbb{E}\left[\mathbb{E}_{s\sim d_\mu^\star}\left[\langle q_s^t - Q_s^t, \pi_s^\star - \pi_s^t\rangle\right]\right] \\ &\overset{(c)}{\geq} \mathbb{E}[\Delta_t](1-\gamma) - \tau(1-\gamma),\end{aligned}$$

where (b) follows from Lemma D.4 and (c) follows similarly to (45), i.e., by applying (42) twice with $(d_\mu^\pi, \pi) = (d_\mu^\star, \pi^\star)$ and $(d_\mu^\pi, \pi) = (d_\mu^\star, \pi^t)$.

Plugging the two bounds in (47), dividing both sides by $(1-\gamma)$ and rearranging, we obtain

$$\mathbb{E}\left[\frac{\mathcal{D}_t^{t+1}}{(1-\gamma)\eta_t} + \nu_{t+1}\left(\Delta_{t+1} - \Delta_t - \tau\right) + \Delta_t\right] \leq \mathbb{E}\left[\frac{\mathcal{D}_t^\star}{(1-\gamma)\eta_t} - \frac{\mathcal{D}_{t+1}^\star}{(1-\gamma)\eta_t}\right] + \tau.$$

From Proposition 4.2, we have that $\Delta_{t+1} - \Delta_t - \tau \leq 0$. Consequently, since $\nu_{t+1} \leq \nu_\mu$ by the definition of ν_μ in Assumption (A3), one can lower bound the left hand side of the above inequality by replacing ν_{t+1} by ν_μ, that is,

$$\mathbb{E}\left[\frac{\mathcal{D}_t^{t+1}}{(1-\gamma)\eta_t} + \nu_\mu\left(\Delta_{t+1} - \Delta_t - \tau\right) + \Delta_t\right] \leq \mathbb{E}\left[\frac{\mathcal{D}_t^\star}{(1-\gamma)\eta_t} - \frac{\mathcal{D}_{t+1}^\star}{(1-\gamma)\eta_t}\right] + \tau,$$

which concludes the proof. □

D.5 Proof of the sublinear convergence analysis

In this section, we derive the sublinear convergence result of Theorem 4.3 with non-decreasing step-size.

Proof. Starting from Proposition D.6

$$\mathbb{E}\left[\frac{\mathcal{D}_t^{t+1}}{(1-\gamma)\eta_t} + \nu_\mu\left(\Delta_{t+1} - \Delta_t\right) + \Delta_t\right] \leq \mathbb{E}\left[\frac{\mathcal{D}_t^\star}{(1-\gamma)\eta_t} - \frac{\mathcal{D}_{t+1}^\star}{(1-\gamma)\eta_t}\right] + (1+\nu_\mu)\tau.$$

If $\eta_t \leq \eta_{t+1}$,

$$\mathbb{E}\left[\frac{\mathcal{D}_t^{t+1}}{(1-\gamma)\eta_t} + \nu_\mu\left(\Delta_{t+1} - \Delta_t\right) + \Delta_t\right] \leq \mathbb{E}\left[\frac{\mathcal{D}_t^\star}{(1-\gamma)\eta_t} - \frac{\mathcal{D}_{t+1}^\star}{(1-\gamma)\eta_{t+1}}\right] + (1+\nu_\mu)\tau. \tag{48}$$

Summing up from 0 to $T-1$ and dropping some positive terms on the left hand side and some negative terms on the right hand side, we have

$$\sum_{t<T}\mathbb{E}\left[\Delta_t\right] \leq \frac{\mathcal{D}_0^\star}{(1-\gamma)\eta_0} + \nu_\mu\Delta_0 + T(1+\nu_\mu)\tau \leq \frac{\mathcal{D}_0^\star}{(1-\gamma)\eta_0} + \frac{\nu_\mu}{1-\gamma} + T(1+\nu_\mu)\tau.$$

Notice that $\Delta_0 \leq \frac{1}{1-\gamma}$ as $r(s,a) \in [0,1]$. By dividing T on both side, we yield the proof of the sublinear convergence

$$V^\star(\mu) - \frac{1}{T}\sum_{t<T}\mathbb{E}\left[V^t(\mu)\right] \leq \frac{1}{T}\left(\frac{\mathcal{D}_0^\star}{(1-\gamma)\eta_0} + \frac{\nu_\mu}{1-\gamma}\right) + (1+\nu_\mu)\tau.$$

□

D.6 Proof of the linear convergence analysis

In this section, we derive the linear convergence result of Theorem 4.3 with exponentially increasing step-size.

Proof. Starting from Proposition D.6 by dropping $\frac{\mathcal{D}_t^{t+1}}{(1-\gamma)\eta_t}$ on the left hand side, we have

$$\mathbb{E}\left[\nu_\mu\left(\Delta_{t+1}-\Delta_t\right)+\Delta_t\right] \leq \mathbb{E}\left[\frac{\mathcal{D}_t^\star}{(1-\gamma)\eta_t}-\frac{\mathcal{D}_{t+1}^\star}{(1-\gamma)\eta_t}\right]+(1+\nu_\mu)\tau.$$

Dividing ν_μ on both side and rearranging, we obtain

$$\mathbb{E}\left[\Delta_{t+1}+\frac{\mathcal{D}_{t+1}^\star}{(1-\gamma)\nu_\mu\eta_t}\right] \leq \left(1-\frac{1}{\nu_\mu}\right)\mathbb{E}\left[\Delta_t+\frac{\mathcal{D}_t^\star}{(1-\gamma)\eta_t(\nu_\mu-1)}\right]+\left(1+\frac{1}{\nu_\mu}\right)\tau.$$

If the step-sizes satisfy $\eta_{t+1}(\nu_\mu-1)\geq\eta_t\nu_\mu$ with $\nu_\mu\geq 1$, then

$$\mathbb{E}\left[\Delta_{t+1}+\frac{\mathcal{D}_{t+1}^\star}{(1-\gamma)\eta_{t+1}(\nu_\mu-1)}\right] \leq \left(1-\frac{1}{\nu_\mu}\right)\mathbb{E}\left[\Delta_t+\frac{\mathcal{D}_t^\star}{(1-\gamma)\eta_t(\nu_\mu-1)}\right]+\left(1+\frac{1}{\nu_\mu}\right)\tau.$$

Now we need the following simple fact, whose proof is straightforward and thus omitted.

Suppose $0<\alpha<1, b>0$ and a nonnegative sequence $\{a_t\}_{t\geq 0}$ satisfies

$$a_{t+1}\leq \alpha a_t+b \qquad \forall t\geq 0.$$

Then for all $t\geq 0$,

$$a_t\leq \alpha^t a_0+\frac{b}{1-\alpha}.$$

The proof of the linear convergence analysis follows by applying this fact with $a_t = \mathbb{E}\left[\Delta_t+\frac{\mathcal{D}_t^\star}{(1-\gamma)\eta_t(\nu_\mu-1)}\right]$, $\alpha=1-\frac{1}{\nu_\mu}$ and $b=\left(1+\frac{1}{\nu_\mu}\right)\tau$. □

E Discussion on Assumption (A1)

Assumption (A1) encodes a form of realizability assumption for the parameterization class $\mathcal{F}^\Theta$, that is, we assume that for all $t\leq T$ there exists a function $f^\theta\in\mathcal{F}^\Theta$ such that

$$\left\|f^\theta-Q^t-\eta_t^{-1}\nabla h(\pi^t)\right\|_{L_2(v_t)}^2\leq\varepsilon_{\text{approx}}.$$

When $\mathcal{F}^\Theta$ is a class of sufficiently large shallow neural networks, this realizability assumption holds as it has been shown that shallow neural networks are universal approximators [Ji et al., 2019]. It is, however, possible to relax Assumption (A1). In particular, the condition

$$\frac{1}{T}\sum_{t<T}\sqrt{\mathbb{E}\left[\mathbb{E}\left\|f^{t+1}-Q^t-\eta_t^{-1}\nabla h(\pi^t)\right\|_{L_2(v_t)}^2\right]}\leq\sqrt{\varepsilon_{\text{approx}}} \tag{49}$$

can replace Assumption (A1) and is sufficient for the sublinear convergence rate in Theorem 4.3 to hold. Equation (49) shows that the realizability assumption does not need to hold for all $t<T$, but only needs to hold on average over T iterations. Similarly, the condition

$$\sum_{t\leq T}\left(1-\frac{1}{\nu_\mu}\right)^{T-t}\frac{1}{\nu_\mu}\sqrt{\mathbb{E}\left[\mathbb{E}\left\|f^{t+1}-Q^t-\eta_t^{-1}\nabla h(\pi^t)\right\|_{L_2(v_t)}^2\right]}\leq\sqrt{\varepsilon_{\text{approx}}} \tag{50}$$

can replace Assumption (A1) and is sufficient for the linear convergence rate in Theorem 4.3 to hold. Additionally, requiring, for all $t<T$,

$$\mathbb{E}\left[\mathbb{E}\left\|f^{t+1}-Q^t-\eta_t^{-1}\nabla h(\pi^t)\right\|_{L_2(v_t)}^2\right]\leq\frac{\nu_\mu^2}{T^2}\left(1-\frac{1}{\nu_\mu}\right)^{-2(T-t)}\varepsilon_{\text{approx}} \tag{51}$$

is sufficient for Equation (50) to hold. Equation (51) shows that the error floor in the linear convergence rate is less influenced by approximation errors made in early iterations, which are discounted by the term $\left(1-\frac{1}{\nu_\mu}\right)$. On the other hand, the realizability assumption becomes relevant once the algorithm approaches convergence, i.e., when $t\simeq T$ and $Q^t\simeq Q^\star$, as the discount term $\left(1-\frac{1}{\nu_\mu}\right)$ is applied fewer times.

F Discussion on the concentrability coefficients and the distribution mismatch coefficients

In our convergence analysis, Assumptions (A2) and (A3) involve the concentrability coefficient C_v and the distribution mismatch coefficient ν_μ, which are potentially large. We give extensive discussions on them, respectively.

Concentrability coefficient C_v. As discussed in Yuan et al. [2023, Appendix H], the issue of having (potentially large) concentrability coefficient (Assumptions (A2)) is unavoidable in all the fast linear convergence analysis of approximate PMD due to the approximation error $\varepsilon_{\text{approx}}$ of the Q-function [Cen et al., 2021, Zhan et al., 2021, Lan, 2022, Cayci et al., 2022, Xiao, 2022, Chen and Theja Maguluri, 2022, Alfano and Rebeschini, 2022, Yuan et al., 2023]. Indeed, in the fast linear convergence analysis of PMD, the concentrability coefficient is always along with the approximation error $\varepsilon_{\text{approx}}$ under the form of $C_v\varepsilon_{\text{approx}}$, which is the case in Theorem 4.3. To not get the concentrability coefficient involved yet maintain the linear convergence of PMD, one needs to consider the exact PMD in the tabular setting [see Xiao, 2022, Theorem 10]. Consequently, the PMD update is deterministic and the full policy space $\Delta(\mathcal{A})^{\mathcal{S}}$ is considered. In this setting, at each time t, it exists θ^{t+1} such that, for any state-action distribution v^t,

$$\left\|f^{t+1} - Q^t - \eta_t^{-1}\nabla h(\pi^t)\right\|_{L_2(v_t)}^2 = 0 = \varepsilon_{\text{approx}},$$

and C_v is ignored in the convergence analysis thanks to the vanishing of $\varepsilon_{\text{approx}}$. Note that the PMD analysis in the seminal paper by Agarwal et al. [2021] does not use such a coefficient, but a condition number instead. The condition number is controllable to be relatively small, so that the error term in their PMD analysis is smaller than ours. However, their PMD analysis has only a sublinear convergence rate, while ours enjoys a fast linear convergence rate. And their proof is quite different from ours. It remains an open question whether one can both avoid using the concentrability coefficient and maintain the linear convergence of PMD.

Now we compare our concentrability coefficient C_v with others used in the fast linear convergence analysis of approximate PMD [Cen et al., 2021, Zhan et al., 2021, Lan, 2022, Cayci et al., 2021, Xiao, 2022, Chen and Theja Maguluri, 2022, Alfano and Rebeschini, 2022, Yuan et al., 2023]. Previously, as discussed in Yuan et al. [2023, Appendix H], their concentrability coefficient C_v was the "best" known one among others in the literature. First, theirs took the weakest assumptions on errors among Lan [2022], Xiao [2022] and Chen and Theja Maguluri [2022] by using the ℓ_2-norm instead of the L_∞ supremum norm over the approximation error $\varepsilon_{\text{approx}}$. Second, it did not impose any restrictions on the MDP dynamics compared to Cayci et al. [2021], as the concentrability coefficient of Yuan et al. [2023] was independent to the iterates.

Indeed, Yuan et al. [2023] choose v^t such that, for all $(s,a) \in \mathcal{S}\times\mathcal{A}$,

$$v^t(s,a) = (1-\gamma)\,\mathbb{E}_{(s_0,a_0)\sim\nu}\left[\sum_{t'=0}^{\infty}\gamma^{t'}P(s_{t'}=s, a_{t'}=a \mid \pi^t, s_0, a_0)\right],$$

where ν is an initial state-action distribution chosen by the user. In this setting, we have

$$v^t(s,a) \geq (1-\gamma)\nu(s,a).$$

From the above lower bound of v^t, we obtain that

$$\begin{aligned}\mathbb{E}_{(s,a)\sim v^t}\left[\left(\frac{d_\mu^\pi(s)\pi(a|s)}{v^t(s,a)}\right)^2\right] &= \int_{(s,a)\in\mathcal{S}\times\mathcal{A}}\frac{d_\mu^\pi(s)^2\pi(a|s)^2}{v^t(s,a)} \\ &\leq \int_{(s,a)\in\mathcal{S}\times\mathcal{A}}\frac{1}{v^t(s,a)} \leq \frac{1}{(1-\gamma)\min_{(s,a)\in\mathcal{S}\times\mathcal{A}}\nu(s,a)},\end{aligned}$$

where the finite upper bound is independent to t.

As mentioned right after Assumption (A2), the assumption on our concentrability coefficient C_v is weaker than the one in Yuan et al. [2023, Assumption 9], as we have the full control over

v^t while Yuan et al. [2023] only has the full control over the initial state-action distribution ν. In particular, our concentrability coefficient C_v recovers the previous best known one in Yuan et al. [2023] as a special case. Consequently, our concentrability coefficient C_v becomes the "best" with the full control over v^t when other concentrability coefficients are infinite or require strong assumptions [Scherrer, 2014].

In general, for the ratio $\mathbb{E}_{(s,a)\sim v^t}\left[\left(\frac{d_\mu^\pi(s)\pi(a|s)}{v^t(s,a)}\right)^2\right]$ to have a finite upper bound C_v, it is important that v^t covers well the state and action spaces so that the upper bound is independent to t. However, the upper bound $\frac{1}{(1-\gamma)\min_{(s,a)\in\mathcal{S}\times\mathcal{A}}\nu(s,a)}$ in Yuan et al. [2023] is very pessimistic. Indeed, when π^t and π^{t+1} converge to $\pi^\star$, one reasonable choice of v^t is to choose $v^t \in \{d_\mu^\star\cdot\pi^\star, d_\mu^{t+1}\cdot\pi^{t+1}, d_\mu^\star\cdot\pi^t, d_\mu^{t+1}\cdot\pi^t\}$ such that C_v is closed to 1.

We also refer to Yuan et al. [2023, Appendix H] for more discussions on the concentrability coefficient.

Distribution mismatch coefficient ν_μ. As mentioned right after (A3), we have that

$$\max_{s\in\mathcal{S}}\frac{d_\mu^\star(s)}{d_\mu^t(s)} \le \frac{1}{1-\gamma}\max_{s\in\mathcal{S}}\frac{d_\mu^\star(s)}{\mu(s)} := \nu'_\mu,$$

which is a sufficient upper bound for ν_μ. As discussed in Yuan et al. [2023, Appendix H],

$$1/(1-\gamma) \le \nu'_\mu \le 1/((1-\gamma)\min_s \mu(s)).$$

The upper bound $1/((1-\gamma)\min_s\mu(s))$ of ν'_μ is very pessimistic and the lower bound $\nu'_\mu = 1/(1-\gamma)$ is often achieved by choosing $\mu = d_\mu^\star$.

Furthermore, if μ does not have full support on the state space, i.e., the upper bound $1/((1-\gamma)\min_s\mu(s))$ might be infinite, one can always convert the convergence guarantees for some state distribution $\mu' \in \Delta(\mathcal{S})$ with full support such that

$$\begin{aligned} V^\star(\mu) - \mathbb{E}[V^T(\mu)] &= \mathbb{E}\left[\int_{s\in\mathcal{S}}\frac{\mu(s)}{\mu'(s)}\mu'(s)\left(V^\star(s) - V^T(s)\right)\right] \\ &\le \max_{s\in\mathcal{S}}\frac{\mu(s)}{\mu'(s)}\left(V^\star(\mu') - \mathbb{E}[V^T(\mu')]\right). \end{aligned}$$

Then by the linear convergence result of Theorem 4.3, we only transfer the original convergence guarantee to $V^\star(\mu') - \mathbb{E}[V^T(\mu')]$ up to a scaling factor $\max_{s\in\mathcal{S}}\frac{\mu(s)}{\mu'(s)}$ with an arbitrary distribution μ' such that ν'_μ is finite.

Finally, if d_μ^t converges to $d_\mu^\star$ which is the case of AMPO through the proof of our Theorem 4.3, then $\max_{s\in\mathcal{S}}\frac{d_\mu^\star(s)}{d_\mu^t(s)}$ converges to 1. This might imply superlinear convergence results as discussed in Xiao [2022, Section 4.3]. In this case, the notion of the distribution mismatch coefficients ν_μ no longer exists for the superlinear convergence analysis.

We also refer to Yuan et al. [2023, Appendix H] for more discussions on the distribution mismatch coefficient.

G Sample complexity for neural network parameterization

We prove here Corollary 4.4 through a result by Allen-Zhu et al. [2019a, Theorem 1 and Example 3.1]. We first give a simplified version of this result and then we show how to use it to prove Corollary 4.4.

Consider learning some unknown distribution $\mathcal{D}$ of data points $z = (x, y) \in \mathbb{R}^d \times \mathcal{Y}$, where x is the input point and y is the label. Without loss of generality, assume $\|x\|_2 = 1$ and $x_d = 1/2$. Consider a loss function $L : \mathbb{R}^k \times \mathcal{Y} \to \mathbb{R}$ such that for every $y \in \mathcal{Y}$, the function $L(\cdot, y)$ is non-negative, convex, 1-Lipschitz continuous and $L(0, y) \in [0, 1]$. This includes both the cross-entropy loss and the ℓ_2-regression loss (for bounded $\mathcal{Y}$).

Let $g : \mathbb{R} \to \mathbb{R}$ be a smooth activation function such that $g(z) = e^z$, $\sin(z)$, $\texttt{sigmoid}(z)$, $\tanh(z)$ or is a low degree polynomial.

Define $F^\star : \mathbb{R}^d \to \mathbb{R}^k$ such that $OPT = \mathbb{E}_{\mathcal{D}}[L(F^\star(x), y)]$ is the smallest population error made by a neural network of the form $F^\star = A^\star g(W^\star x)$, where $A^\star \in \mathbb{R}^{k \times p}$ and $W^\star \in \mathbb{R}^{p \times d}$. Assume for simplicity that the rows of W^* have ℓ_2-norm 1 and each element of A^* is less or equal than 1.

Define a ReLU neural network $F(x, W_0) = A_0 \sigma(W_0 x + b_0)$, where $A_0 \in \mathbb{R}^{k \times m}$, $W_0 \in \mathbb{R}^{m \times d}$, the entries of W_0 and b_0 are i.i.d. random Gaussians from $\mathcal{N}(0, 1/m)$ and the entries of A are i.i.d. random Gaussians from $\mathcal{N}(0, \varepsilon_A)$, for $\varepsilon_A \in (0, 1]$. We train the weights W of this neural network through stochastic gradient descent over a dataset with N i.i.d. samples from $\mathcal{D}$, i.e., we update $W_{t+1} = W_t - \eta g_t$, where $\mathbb{E}[g_t] = \nabla \mathbb{E}_{\mathcal{D}}[L(F(x, W_0 + W_t), y)]$.

Theorem G.1 (Theorem 1 of Allen-Zhu et al. [2019a]). *Let $\varepsilon \in (0, O(1/pk))$, choose $\varepsilon_A = \varepsilon / \widetilde{\Theta}(1)$ for the initialization and learning rate $\eta = \widetilde{\Theta}\left(\frac{1}{\varepsilon k m}\right)$. SGD finds a set of parameters such that*

$$\frac{1}{J} \sum_{n=0}^{J-1} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[L\left(F(x; W^{(0)} + W_t), y \right) \right] \leq OPT + \varepsilon$$

with probability $1 - e^{-c \log^2 m}$ over the random initialization, for a sufficiently large constant c, with

$$\textit{size } m = \frac{\texttt{poly}(k, p)}{\texttt{poly}(\varepsilon)} \textit{ and sample complexity } \min\{N, J\} = \frac{\texttt{poly}(k, p, \log m)}{\varepsilon^2}.$$

Theorem G.1 shows that it is possible to achieve the population error OPT by training a two-layer ReLU network with SGD, and quantifies the number of samples needed to do so.

We make the following assumption to address the population error in our setting.

Assumption G.2. Let $g : \mathbb{R} \to \mathbb{R}$ be a smooth activation function such that $g(z) = e^z$, $\sin(z)$, $\texttt{sigmoid}(z)$, $\tanh(z)$ or is a low degree polynomial. For all time-steps t, we assume that there exists a target network $F^{\star,t} : \mathbb{R}^d \to \mathbb{R}^k$, with

$$F^{\star,t} = (f_1^{\star,t}, \ldots, f_k^{\star,t}) \quad \text{and} \quad f_r^{\star,t}(x) = \sum_{i=1}^{p} a_{r,i}^{\star,t} g(\langle w_{1,i}^{\star,t}, x \rangle) \langle w_{2,i}^{\star,t}, x \rangle$$

where $w_{1,i}^{\star,t} \in \mathbb{R}^d$, $w_{2,i}^{\star,t} \in \mathbb{R}^d$, and $a_{r,i}^{\star,t} \in \mathbb{R}$, such that

$$\mathbb{E}\left[\left\| F^{\star,t} - Q^t - \eta_t^{-1} \nabla h(\pi^t) \right\|_{L_2(\nu_t)}^2 \right] \leq OPT.$$

We assume for simplicity $\|w_{1,i}^{\star,t}\|_2 = \|w_{2,i}^{\star,t}\|_2 = 1$ and $|a_{r,i}^{\star,t}| \leq 1$.

Assumptions similar to Assumption G.2 have already been made in the literature, such as the bias assumption in the compatible function approximation framework studied by Agarwal et al. [2021]. The term OPT represents the minimum error incurred by a target network parametrized as $F^{\star,t}$ when solving the regression problem in Line 1 of Algorithm 1.

We are now ready to prove Corollary 4.4, which uses Algorithm 4 to obtain an unbiased estimate of the current Q-function. We assume to be in the same setting as Theorem G.1

Proof of Corollary 4.4. We aim to find a policy π^T such that

$$V^\star(\mu) - \mathbb{E}\left[V^T(\mu)\right] \leq \varepsilon. \tag{52}$$

Suppose the total number of iterations, that is policy optimization steps, in AMPO is T. We need the bound in Assumption (A1) to hold for all T with probability $1 - e^{-c \log^2 m}$, which means that at each iteration the bound should hold with probability $1 - T^{-1} e^{-c \log^2 m}$. Through Algorithm 4, the expected number of samples needed to obtain an unbiased estimate

Algorithm 4: Sampler for an unbiased estimate $\widehat{Q^t}(s,a)$ of $Q^t(s,a)$

Input: Initial state-action couple (s_0, a_0), policy π^t, discount factor $\gamma \in [0,1)$

Initialize $\widehat{Q^t}(s_0,a_0) = r(s_0,a_0)$, the time step $n=0$.
while *True* **do**
  With probability γ**:**
    Sample $s_{n+1} \sim P(\cdot \mid s_n, a_n)$
    Sample $a_{n+1} \sim \pi^t(\cdot|s_{n+1})$
    $\widehat{Q^t}(s_0,a_0) \leftarrow \widehat{Q^t}(s_0,a_0) + r(s_{n+1},a_{n+1})$
    $n \leftarrow n+1$
  Otherwise with probability $(1-\gamma)$**:**
    break ▷ Accept $\widehat{Q}_{s_h,a_h}(\theta)$

Output: $\widehat{Q^t}(s_0,a_0)$

of the current Q-function is $(1-\gamma)^{-1}$. Therefore, using Theorem G.1, at each iteration of AMPO we need at most

$$\frac{\texttt{poly}(k,p,\log m,\log T)}{\varepsilon_{\text{approx}}^2(1-\gamma)}$$

samples for SGD to find parameters that satisfy Assumption (A1) with probability $1 - T^{-1}e^{-c\log^2 m}$. To obtain (52), we need

$$\frac{1}{1-\gamma}\left(1-\frac{1}{\nu_\mu}\right)^T\left(1+\frac{\mathcal{D}_0^\star}{\eta_0(\nu_\mu-1)}\right) \le \frac{\varepsilon}{2} \quad \text{and} \quad \frac{2(1+\nu_\mu)\sqrt{C_v\varepsilon_{\text{approx}}}}{1-\gamma} \le \frac{\varepsilon}{2}. \tag{53}$$

Solving for T and $\varepsilon_{\text{approx}}$ and multiplying them together, we obtain the sample complexity of AMPO, that is

$$\widetilde{\mathcal{O}}\left(\frac{\texttt{poly}(k,p,\log m)C_v^2\nu_\mu^5}{\varepsilon^4(1-\gamma)^6}\right).$$

Due to the statement of Theorem G.1, we cannot guarantee the approximation error incurred by the learner network to be smaller than OPT. Consequently, we have that

$$\varepsilon \ge \frac{4(1+\nu_\mu)\sqrt{C_v OPT}}{1-\gamma}.$$

A similar bound can be applied to any proof that contains the bias assumption introduced by [1]. □

We can obtain an improvement over Corollary 4.4 by using the relaxed assumptions in Appendix E, in particular using the condition in (51).

Corollary G.3. *In the setting of Theorem 4.3, replace Assumption (A1) with the condition*

$$\mathbb{E}\left[\mathbb{E}\left\|f^{t+1} - Q^t - \eta_t^{-1}\nabla h(\pi^t)\right\|_{L_2(v_t)}^2\right] \le \frac{\nu_\mu^2}{T^2}\left(1-\frac{1}{\nu_\mu}\right)^{-2(T-t)}\varepsilon_{\text{approx}}, \tag{54}$$

for all $t < T$. Let the parameterization class $\mathcal{F}^\Theta$ consist of sufficiently wide shallow ReLU neural networks. Using an exponentially increasing step-size and solving the minimization problem in Line 1 with SGD as in (17), the number of samples required by AMPO to find an ε-optimal policy with high probability is $\widetilde{\Theta}(C_v^2\nu_\mu^4/\varepsilon^4(1-\gamma)^6)$.

Proof. The proof follow that of Corollary 4.4. Using Theorem G.1, at each iteration t of AMPO, we need at most

$$\frac{T^2}{\nu_\mu^2}\left(1-\frac{1}{\nu_\mu}\right)^{2(T-t)}\frac{\texttt{poly}(k,p,\log m,\log T)}{\varepsilon_{\text{approx}}^2(1-\gamma)}$$

samples for SGD to find parameters that satisfy condition (54) with probability $1 - T^{-1}e^{-c\log^2 m}$. Summing over T total iterations of AMPO we obtain that the total number of samples needed is

$$\begin{aligned}
&\sum_{t\leq T}\frac{T^2}{\nu_\mu^2}\left(1-\frac{1}{\nu_\mu}\right)^{2(T-t)}\frac{\texttt{poly}(k,p,\log m,\log T)}{\varepsilon^2_{\text{approx}}(1-\gamma)}\\
&\quad=\frac{T^2}{\nu_\mu^2}\left(1-\frac{1}{\nu_\mu}\right)^{2T}\frac{\texttt{poly}(k,p,\log m,\log T)}{\varepsilon^2_{\text{approx}}(1-\gamma)}\sum_{t\leq T}\left(1-\frac{1}{\nu_\mu}\right)^{-2t}\\
&\quad=\frac{T^2}{\nu_\mu^2}\left(1-\frac{1}{\nu_\mu}\right)^{2T}\frac{\texttt{poly}(k,p,\log m,\log T)}{\varepsilon^2_{\text{approx}}(1-\gamma)}\frac{\left(\left(1-\frac{1}{\nu_\mu}\right)^{-2(T+1)}-1\right)}{\left(\left(\frac{1}{1-\nu_\mu}\right)^{-2}-1\right)}\\
&\quad\leq\mathcal{O}\left(\frac{T^2}{\nu_\mu^2}\frac{\texttt{poly}(k,p,\log m,\log T)}{\varepsilon^2_{\text{approx}}(1-\gamma)}\right)
\end{aligned}$$

Replacing T and $\varepsilon_{\text{approx}}$ with the solutions of (53) gives the result. □

At this stage, it is important to note that choosing a method different from the one proposed by Allen-Zhu et al. [2019b] to solve Line 1 in Algorithm 1 of our paper with neural networks can lead to alternative, and possibly better, sample complexity results for AMPO. For example, we can obtain a sample complexity result for AMPO that does not involve a target network using results from Ji et al. [2019] and Cayci et al. [2022], although this requires introducing more notation and background results compared to Corollary 4.4 (since in Cayci et al. [2022] they employ a temporal-difference-based algorithm, that is Algorithm 3 in their work, to obtain a neural network estimate $\widehat{Q}^t$ of Q^t, while in Ji et al. [2019] they provide a method based on Fourier transforms to approximate a target function through shallow ReLU networks). We outline below the steps in order to do so (and additional details including the precise statements of the results we use and how we use them are provided thereafter for the sake of completeness).

Step 1) We first split the approximation error in Assumption (A1) into a critic error $\mathbb{E}[\sqrt{\|\widehat{Q}^t - Q^t\|^2_{L_2(v^t)}}] \leq \varepsilon_{\text{critic}}$ and an actor error $\mathbb{E}[\sqrt{\|f^{t+1} - \widehat{Q}^t - \eta_t^{-1}\nabla h(\pi^t)\|^2_{L_2(v^t)}}] \leq \varepsilon_{\text{actor}}$. In this case, the linear convergence rate in our Theorem 4.3 becomes

$$V^\star(\mu) - \mathbb{E}\left[V^T(\mu)\right] \leq \frac{1}{1-\gamma}\left(1-\frac{1}{\nu_\mu}\right)^T\left(1+\frac{\mathcal{D}_0^\star}{\eta_0(\nu_\mu-1)}\right) + \frac{2(1+\nu_\mu)\sqrt{C_v}(\varepsilon_{\text{critic}}+\varepsilon_{\text{actor}})}{1-\gamma}.$$

[We can obtain this alternative statement by modifying the passages in Appendix D.2. In particular, writing $f^{t+1} - Q^t - \eta_t^{-1}\nabla h(\pi^t) = (f^{t+1} - \widehat{Q}^t - \eta_t^{-1}\nabla h(\pi^t)) + (Q^t - \widehat{Q}^t)$ and bounding the two terms with the same procedure in Appendix D.2 leads to this alternative expression for the error.]

We will next deal with the critic error and actor error separately.

Step 2) Critic error. Under a realizability assumption that we provide below along with the statement of the theorem (Assumption 2 in Cayci et al. [2022]), Theorem 1 from Cayci et al. [2022] gives that the sample complexity required to obtain $\mathbb{E}[\sqrt{\|\widehat{Q}^t - Q^t\|^2_{L_2(d_\mu^t\cdot\pi^t)}}] \leq \varepsilon$ is $\widetilde{O}(\varepsilon^{-4}(1-\gamma)^{-2})$, while the required network width is $\widetilde{O}(\varepsilon^{-2})$.

Step 3) Actor error. Using Theorem E.1 from Ji et al. [2019], we obtain that $\mathbb{E}[\sqrt{\|f^{t+1} - \widehat{Q}^t - \eta_t^{-1}\nabla h(\pi^t)\|^2_{L_2(v^t)}}]$ can be made arbitrarily small by tuning the width of f^{t+1}, without using further samples.

Step 4) Replacing Equation (53) with the sample complexity of the critic, we obtain that the sample complexity of AMPO is $\widetilde{O}(C_v^2\nu_\mu^5/\varepsilon^4(1-\gamma)^7)$, which does not depend on the error made by a target network.

To the best of our knowledge, this result improves upon the previous best result on the sample complexity of a PG method with neural network parameterization [Cayci et al., 2022], i.e., $\widetilde{\mathcal{O}}(C_v^2/\varepsilon^6(1-\gamma)^9)$.

We now provide the statements of the aforementioned results we used.

Recalling Theorem 1 in Cayci et al. [2022] and its assumptions. Consider the following space of mappings:

$$\mathcal{H}_{\bar{\nu}} = \{v : \mathbb{R}^d \to \mathbb{R}^d : \sup_{w\in\mathbb{R}^d} \|v(w)\|_2 \leq \bar{\nu}\},$$

and the function class:

$$\mathcal{F}_{\bar{\nu}} = \Big\{g(\cdot) = \mathbb{E}_{w_0\sim\mathcal{N}(0,I_d)}[\langle v(w_0),\cdot\rangle \mathbb{I}\{\langle w_0,\cdot\rangle > 0\}] : v \in \mathcal{H}_{\bar{\nu}}\Big\}.$$

Consider the following realizability assumption for the Q-function.

Assumption G.4 (Assumption 2 in Cayci et al. [2022])**.** For any $t \geq 0$, we assume that $Q^t \in \mathcal{F}_{\bar{\nu}}$ for some $\bar{\nu} > 0$.

Theorem G.5 (Theorem 1 in Cayci et al. [2022])**.** *Under Assumption 2 in Cayci et al. [2022], for any error probability $\delta \in (0,1)$, let*

$$\ell(m',\delta) = 4\sqrt{\log(2m'+1)} + 4\sqrt{\log(T/\delta)},$$

and $R > \bar{\nu}$. Then, for any target error $\varepsilon > 0$, number of iterations $T' \in \mathbb{N}$, network width

$$m' > \frac{16\Big(\bar{\nu} + \big(R + \ell(m',\delta)\big)\big(\bar{\nu}+R\big)\Big)^2}{(1-\gamma)^2\varepsilon^2},$$

and step-size

$$\alpha_C = \frac{\varepsilon^2(1-\gamma)}{(1+2R)^2},$$

Algorithm 3 in Cayci et al. [2022] yields the following bound:

$$\mathbb{E}\Big[\sqrt{\|\widehat{Q}^t - Q^t\|^2_{L_2(d_\mu^t\cdot\pi^t)}}\mathbb{I}_{A_2}\Big] \leq \frac{(1+2R)\bar{\nu}}{\varepsilon(1-\gamma)\sqrt{T'}} + 3\varepsilon,$$

where A_2 holds with probability at least $1-\delta$ over the random initializations of the critic network $\widehat{Q}^t$.

As indicated in Cayci et al. [2022], a consequence of this result is that in order to achieve a target error less than $\varepsilon > 0$, a network width of $m' = \widetilde{O}\Big(\frac{\bar{\nu}^4}{\varepsilon^2}\Big)$ and iteration complexity $O\Big(\frac{(1+2\bar{\nu})^2\bar{\nu}^2}{(1-\gamma)^2\varepsilon^4}\Big)$ suffice.

The statement of Theorem 1 in Cayci et al. [2022] can be readily applied to obtain the sample complexity of the critic.

Recalling Theorem E.1 in [3] and its assumptions Let $g : \mathbb{R}^n \to \mathbb{R}$ be given and define the modulus of continuity ω_g as

$$\omega_g(\delta) := \sup_{x,x'\in\mathbb{R}^n} \{g(x) - g(x') : \max(\|x\|_2, \|x'\|_2) \leq 1+\delta, \|x-x'\|_2 \leq \delta\}.$$

If g is continuous, then ω_g is not only finite for all inputs, but moreover $\lim_{\delta\to 0}\omega_g(\delta) \to 0$.

Denote $\|p\|_{L_1} = \int |p(w)|dw$. Define a sample from a signed density $p : \mathbb{R}^{n+1} \to \mathbb{R}$ with $\|p\|_{L_1} < \infty$ as (w,b,s), where $(w,b) \in \mathbb{R}$ is sampled from the probability density $|p|/\|p\|_{L_1}$ and $s = sign(p(w,b))$

Theorem G.6 (Theorem E.1 in Ji et al. [2019])**.** *Let $g : \mathbb{R}^n \to \mathbb{R}$, $\delta > 0$ and $\omega_g(\delta)$ be as above and define for $x \in \mathbb{R}^n$*

$$M := \sup_{\|x\|\leq 1+\delta} |g(x)|, \qquad g_{|\delta}(x) = f(x)\mathbb{I}[\|x\| \leq 1+\delta], \qquad \alpha := \frac{\delta}{\sqrt{\delta} + \sqrt{2\log(2M/\omega_g(\delta))}}.$$

Let G_α be a gaussian distribution on $\mathbb{R}^n$ with mean 0 and variance $\alpha^2\mathcal{I}$. Define the Gaussian convolution $l = g_{|\delta} * G_\alpha$ with Fourier transform $\widehat{l}$ satisfying radial decomposition $\widehat{l}(w) = |\widehat{l}(w)| \exp(2\pi i \theta_h(w))$. Let P be a probability distribution supported on $\|x\| \leq 1$. Additionally define

$$
\begin{aligned}
c &:= g(0)g(0) \int |\widehat{l}(w)| \big[\cos(2\pi(\theta_l(w) - \|w\|_2)) - 2\pi\|w\|_2 \sin(2\pi(\theta_l(w) - \|w\|_2)) \big] dw \\
a &= \int w|\widehat{l}(w)|dw \\
r &= \sqrt{n} + 2\sqrt{\log \frac{24\pi^2(\sqrt{d}+7)^2\|g_{|\delta}\|_{L_1}}{\omega_g(\delta)}} \\
p &:= 4\pi^2|\widehat{l}(w)| \cos(2\pi(\|w\|_2 - b))\mathbb{I}[|b| \leq \|w\| \leq r],
\end{aligned}
$$

and for convenience create fake (weight, bias, sign) triples

$$(w,b,s)_{m+1} := (0, c, m\, sign(c)), \quad (w,b,s)_{m+2} := (a, 0, m), \quad (w,b,s)_{m+3} := (-a, 0, -m).$$

Then

$$
\begin{aligned}
|c| &\leq M + 2\sqrt{n}\|g_{|\delta}\|_{L_1}(2\pi\alpha^2)^{-d/2}, \\
\|p\|_{L_1} &\leq 2\|g_{|\delta}\|_{L_1}\sqrt{\frac{(2\pi)^3 n}{(2\pi\alpha^2)^{n+1}}},
\end{aligned}
$$

and with probability at least $1 - 3\lambda$ over a draw of $((s_j, w_j, b_j))_{j=1}^m$ from p

$$\sqrt{\left\| g - \frac{1}{m}\sum_{j=1}^{m+3} s_j \sigma(\langle w_j, x\rangle + b_j) \right\|_{L(P)}} \leq 3\omega_g(\delta) + \frac{r\|p\|_{L_1}}{\sqrt{m}} \left[1 + \sqrt{2\log(1/\lambda)}\right].$$

We can then characterize the error of the actor by choosing $x = (s, a)$, $g = \widehat{Q}^t + \eta_t^{-1}\nabla h(\pi^t)$, and $f^{t+1} = \frac{1}{m}\sum_{j=1}^{m+3} s_j \sigma(\langle w_j, x\rangle + b_j)$. We can then make the actor error arbitrarily small by tuning the network width m and δ (note that, since both $\widehat{Q}^t$ and f^t are continuous neural networks, g is a continuous function).