A functional mirror ascent view of policy gradient methods with function approximation

Sharan Vaswani [1] **Olivier Bachem** [2] **Simone Totaro** [3] **Robert Müller** [4] **Matthieu Geist** [2] **Marlos Machado** [5] **Pablo Samuel Castro** [2] **Nicolas Le Roux** [6]

Abstract

We use functional mirror ascent to propose a general framework (referred to as FMA-PG) for designing policy gradient methods. The functional perspective distinguishes between a policy's functional representation (what are its sufficient statistics) and its parameterization (how are these statistics represented), and naturally results in computationally efficient off-policy updates. For simple policy parameterizations, the FMA-PG framework ensures that the optimal policy is a fixed point of the updates. It also allows us to handle complex policy parameterizations (e.g., neural networks) while guaranteeing policy improvement. Our framework unifies several PG methods and opens the way for designing sample-efficient variants of existing methods. Moreover, it recovers important implementation heuristics (e.g., using forward vs reverse KL divergence) in a principled way. With a softmax functional representation, FMA-PG results in a variant of TRPO with additional desirable properties. It also suggests an improved variant of PPO, whose robustness we empirically demonstrate on MuJoCo.

1 Introduction

Policy gradient (PG) methods [33; 30; 18; 16] are an important class of model-free methods in reinforcement learning. They allow for a differentiable policy parameterization, and can easily handle function approximation and structured state-action spaces. PG methods based on REINFORCE [34] are equipped with strong theoretical guarantees in restricted settings [2; 22; 6]. However, these methods are computationally intractable especially when used with rich function approximation models like neural networks. On the other hand, methods such as TRPO [25], PPO [26] and MPO [1] are efficiently implementable and have good empirical performance [7]. Consequently, these methods are commonly used in deep reinforcement learning [9]. However, they only have weak theoretical guarantees in the tabular setting [15; 25; 23; 10; 27]. Consequently, there are numerous discrepancies between the theory and practice of these methods [19; 21; 8]. Most importantly, there is no principled way to design such PG methods or a unified framework to analyze their theoretical properties.

To address these issues, we view PG methods through the lens of a functional mirror ascent framework. This viewpoint distinguishes between a policy's functional representation (its sufficient statistics) such as the conditional distribution over actions given states or the induced stationary distribution; and its parameterization (how are these statistics represented) such as a linear model or deep network. Our framework unifies different perspectives and provides a principled way to develop and analyze PG methods. In particular, we make the following contributions.

Functional mirror ascent for policy gradient: In Section 3, we specify the functional mirror ascent (FMA) update and connect it to policy gradient methods. In particular, we show that FMA can be interpreted as the repeated application of a policy improvement and a projection (onto the set of feasible policies) operator [11]. For simple policy parameterizations, we prove that the FMA updates are consistent [11] ensuring that the optimal policy (in the class) is a fixed point of the resulting PG method.

Instantiating the FMA framework: In Section 4, we instantiate the general FMA framework with two common functional representations – direct and softmax representations. In the tabular finite state-action setting, we show that the resulting FMA updates recover conservative policy iteration [15] and REINFORCE-based methods [2; 22; 6].

Generic policy gradient framework: In Section 5, we propose a reparameterization technique to handle arbitrarily complex policy parameterizations. This results in FMA-PG, a generic policy gradient framework based on FMA. FMA-PG naturally results in computationally efficient off-

[1]University of Alberta [2]Google Brain [3]Mila, Université de Montréal [4]TU Munich [5]DeepMind [6]Mila, Canada CIFAR AI Chair. Correspondence to: Sharan Vaswani <vaswani.sharan@gmail.com>.

Proceedings of the 38th *International Conference on Machine Learning*, PMLR 139, 2021. Copyright 2021 by the author(s).

policy updates and is instantiated by choosing a functional representation and a policy parameterization. When instantiated with the softmax functional representation, FMA-PG results in an improved, more stable variant of TRPO [25] and MDPO [32]. Moreover, it recovers implementation heuristics (e.g. using forward vs reverse KL divergence) in a principled manner. In Appendix B, we show that the FMA-PG framework can handle stochastic value gradients [12].

Theoretical guarantees: In Section 6, we give a principled way to set the FMA step-size for the direct and softmax representations. With appropriate step-sizes, FMA-PG is guaranteed to improve the policy and converge to a stationary point for *any arbitrary policy parameterization*.

Experimental evaluation: FMA-PG suggests a variant of PPO [26], whose robustness and efficiency we demonstrate on the MuJoco environment [31] (Appendix A).

2 Problem Formulation

We consider an infinite-horizon discounted Markov decision process (MDP) [24] defined by the tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, p, r, d_0, \gamma \rangle$ where $\mathcal{S}$ is a potentially infinite set of states, $\mathcal{A}$ is a potentially infinite action set, $p : \mathcal{S} \times \mathcal{A} \to \Delta^{\mathcal{S}}$ is the transition probability function, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, d_0 is the initial distribution of states, and $\gamma \in [0, 1)$ is the discount factor.

Throughout this paper, we distinguish a policy's functional representation from its parameterization. We will use π to denote the *functional representation* of the corresponding policy. A policy's functional representation defines its sufficient statistics and can be non-parametric. For example, we may define a policy via a distribution $p^\pi(\cdot|s)$ over the actions for each state $s \in \mathcal{S}$, which we call the *direct representation*. Such a representation is used for stochastic policies typically used with policy gradient algorithms [29]. Since $p^\pi(\cdot|s)$ is a probability distribution, an equivalent form is the *softmax representation* $p^\pi(a|s) = \exp(z^\pi(a,s)) / \sum_{a'} \exp(z^\pi(a',s))$ where the policy is specified by the $z^\pi(a, s)$ variables. Note that though the direct and softmax representations are equivalent in the class of policies they define, they result in different functional updates (Sections 4.1 and 4.2). The functional representation affects the final algorithm but it is never made explicit.

Regardless of its representation, each policy π induces a distribution $p^\pi(\cdot|s)$ over actions for each state s. It also induces a measure d^π over states such that $d^\pi(s) = \sum_{t=0}^{\infty} \gamma^t \mathbb{P}(s_{t+1} = s \mid s_0 \sim d_0, a_t \sim p^\pi(a_t|s_t))$. Similarly we define μ^π as the unnormalized distribution over state-action pairs induced by policy π, implying that $\mu^\pi(s, a) = d^\pi(s) p^\pi(a|s)$ and $d^\pi(s) = \sum_a \mu^\pi(s, a)$. The expected discounted return for π is defined as $J(\pi) = \mathbb{E}_{s_0, a_0, \ldots} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right]$, where $s_0 \sim d_0, a_t \sim p^\pi(a_t|s_t)$, and $s_{t+1} \sim p(s_{t+1}|s_t, a_t)$. Given a policy representation, the agent's objective is to learn the policy that maximizes the expected discounted return.

While the functional representation defines a policy's sufficient statistics, the parameterization specifies the practical realization of these statistics. The policy parameterization is independent of its functional representation, it is explicit and determined by a model whose parameters θ are optimized by a policy gradient algorithm. For example, we could represent a policy by its state-action occupancy measure and use a linear parameterization to define this measure, implying $\mu^\pi(s, a|\theta) = \langle \theta, \phi(s, a) \rangle$, where θ is the parameter to be optimized and $\phi(s, a)$ are the known features providing information about the state-occupancy measures. Similarly, we could use a neural-network parameterization for the variables that define a policy in its softmax representation, rewriting $z^\pi(a, s) = z^\pi(a, s|\theta)$. The *tabular parameterization* is special and makes the functional representation of a policy equivalent to its parameterization. For a finite state-action MDP with S states and A actions, choosing a tabular parameterization with the softmax representation results in $\theta \in \mathbb{R}^{SA}$ such that $\forall s \in \mathcal{S}, a \in \mathcal{A}, z^\pi(a, s|\theta) = \theta_{s,a}$. The tabular parameterization is studied in previous work [2; 22]. The policy parameterization defines the set Π of realizable (representable) policies, e.g., when using the direct functional representation with a tabular parameterization, the set Π is a simplex. We denote by $\pi^* := \arg\max_{\pi \in \Pi} J(\pi)$ as the *optimal policy* in the class.

3 Functional mirror ascent framework

To specify the functional mirror ascent (FMA) update, we define a strictly convex, differentiable function ϕ as the mirror map. We denote by $D_\phi(\pi, \mu)$ the Bregman divergence associated with the mirror map ϕ between policies π and μ. Each iteration $t \in [T]$ of FMA consists of the update and projection steps [5]: Eq. (1) computes the gradient $\nabla_\pi J(\pi_t)$ with respect to the policy's functional representation and updates π_t to $\pi_{t+1/2}$ using a step-size η; Eq. (2) computes the Bregman projection of $\pi_{t+1/2}$ onto the class of realizable policies, obtaining π_{t+1}.

$$\pi_{t+1/2} = (\nabla\phi)^{-1} \left(\nabla\phi(\pi_t) + \eta \nabla J(\pi_t) \right), \tag{1}$$

$$\pi_{t+1} = \arg\min_{\pi \in \Pi} D_\phi(\pi, \pi_{t+1/2}). \tag{2}$$

The above FMA updates can also be written as [c.f. 5]

$$\pi_{t+1} = \arg\max_{\pi \in \Pi} \left[\langle \pi, \nabla_\pi J(\pi_t) \rangle - \frac{1}{\eta} D_\phi(\pi, \pi_t) \right]. \tag{3}$$

Note that the FMA update is solely in the functional space and requires solving the above projection as a sub-problem. The policy parameterization defines the set Π of realizable policies and influences the difficulty of solving Eq. (3). We now connect the FMA update to policy gradient and explain the conditions under which π^* is its fixed point.

3.1 Connecting FMA to policy gradient

Several iterative PG methods can be viewed as the repeated application of the *improvement* and *projection* operators [11]. Specifically, at iteration t of a PG method (1) An improvement operator $\mathcal{I}$ transforms the current policy π_t into an improved policy $\pi_{t+1/2} = \mathcal{I}\pi_t$. The improvement operator guarantees a higher expected return, implying that $J(\pi_{t+1/2}) \geq J(\pi_t)$. However, the policy $\pi_{t+1/2}$ might be outside of the policy class Π. (2) The projection operator $\mathcal{P}$ projects $\pi_{t+1/2}$ onto set Π to yield $\pi_{t+1} = \mathcal{P}\pi_{t+1/2} = \mathcal{P} \circ \mathcal{I}\pi_t$.

Ghosh et al. [11] introduced the notion of *consistency* of a pair of operators $(\mathcal{I}, \mathcal{P})$ to mean that π^* is a fixed point of $\mathcal{P} \circ \mathcal{I}$. Using a pair of consistent operators is a necessary but not a sufficient condition for convergence to the optimal policy. In this paper, we associate the pair $(\mathcal{I}, \mathcal{P})$ of operators with the update and projection steps of the functional mirror descent step. Specifically, we define the operators $\mathcal{I}$ and $\mathcal{P}$ such that, at iteration t, $\mathcal{I}\pi_t := \pi_{t+1/2}$ (Eq. (1)) and $\mathcal{P} \circ \mathcal{I}\pi_t := \mathcal{P}\pi_{t+1/2} = \pi_{t+1}$ (Eq. (2)). We now show that this choice always results in pairs of consistent operators.

Proposition 1 (Operator consistency for the FMA update)**.** *By defining the improvement and projection operators as the update and projection step of FMA, for the same mirror map (as in Eqs.* (1) *and* (2)*),* π^* *is a fixed point of* $\mathcal{P} \circ \mathcal{I}$*.*

Note that the proof of the above proposition relies on exactly solving the minimization step in Eq. (2). When using the tabular or linear parameterization, the set Π of realizable policies is convex and, since the function $D_\phi(\cdot, \pi)$ is convex for all π, the minimization can be done exactly. In this case, Proposition 1 implies that the optimal policy π^* is a fixed point (of many) of the FMA update.

4 Instantiating the FMA framework

We use FMA with two common functional representations – the direct representation (Section 4.1) and the softmax representation (Section 4.2). In this section, we only consider the functional aspect and the tabular parameterization, while we handle general policy parameterization in Section 5.

4.1 Direct functional representation

In the direct functional representation, the policy π is represented by the set of distributions $p^\pi(\cdot|s)$ over actions for each state $s \in \mathcal{S}$. In this case, $\frac{\partial J(\pi)}{\partial p^\pi(a|s)} = d^\pi(s) Q^\pi(s,a)$. Since $p^\pi(\cdot|s)$ is a set of distributions, we define the mirror map as $\phi(\pi) = \sum_{s\in\mathcal{S}} w(s)\, \phi(p^\pi(\cdot|s))$, where $w(s)$ is any positive weighting on the states s. Note that the positive weights ensure that $\phi(\pi)$ is a valid mirror-map. The resulting Bregman divergence is $D_\phi(\pi, \pi') = \sum_s w(s) D_\phi(p^\pi(\cdot|s), p^{\pi'}(\cdot|s))$, that is, the weighted sum of the Bregman divergences between the action distributions in state s. By choosing $w(s)$ equal to $d^{\pi_t}(s)$, Eq. (3) involves maximizing (over the set Π) $\mathbb{E}_{(s,a)\sim\mu^{\pi_t}}\left[\left(Q^{\pi_t}(s,a)\, \frac{p^\pi(a|s)}{p^{\pi_t}(a|s)}\right) - \frac{1}{\eta} D_\phi(p^\pi(\cdot|s), p^{\pi_t}(\cdot|s))\right]$.

For finite states and actions, the tabular parameterization results in an equivalence between the functional and parametric spaces and the first term is the same as in conservative policy iteration (CPI) [15]. In this case, the feasible set Π is the SA-dimensional simplex. With the squared Euclidean distance as the mirror map, the above equation is the same as the standard REINFORCE update [34; 2]. Choosing the negative entropy as the mirror map results in a Bregman divergence equal to the KL divergence. With this choice and a tabular parameterization, we recover the natural policy gradient update [14; 17]. In this case, the FMA update is also similar to that of uniform TRPO [27] and MDMPI [10].

4.2 Softmax functional representation

Using the softmax functional representation results in an FMA update on the logits $z^\pi(a,s)$ of the conditional distributions $p^\pi(a|s)$. Formally, $p^\pi(a|s) = \frac{\exp(z^\pi(a,s))}{\sum_{a'} \exp(z^\pi(a',s))}$ and the policy gradient theorem yields $\frac{\partial J(\pi)}{\partial z^\pi(a,s)} = d^\pi(s) A^\pi(s,a) p^\pi(a|s)$. Here, $A^\pi(s,a)$ is the advantage function equal to $Q^\pi(s,a) - V^\pi(s)$. Similar to Section 4.1, we use a mirror map $\phi_z(z)$ that decomposes across states, i.e. $\phi_z(z) = \sum_s w(s)\, \phi_z(z^\pi(\cdot,s))$ for some positive weighting w. We denote the corresponding Bregman divergence as D_{ϕ_z}. Using $w(s) = d^{\pi_t}(s)$, Eq. (3) involves maximizing $E_{(s,a)\sim\mu^{\pi_t}}\left[A^{\pi_t}(s,a)\, p^{\pi_t}(a|s) - \frac{1}{\eta}\sum_s w(s) D_{\phi_z}\left(z^\pi(\cdot,s), z^{\pi_t}(\cdot,s)\right)\right]$. In the tabular finite state-action case, with the squared Euclidean mirror map, the update is equal to that analyzed in [2; 22].

A possible choice for ϕ is the normalized exponential, i.e. $\phi_z(z) = \sum_s w(s) \frac{\sum_a \exp(z^\pi(a,s))}{\sum_a \exp(z^{\pi_t}(a,s))}$. With $w(s) = d^{\pi_t}(s)$, Eq. (3) is equal to $\pi_{t+1} = \arg\max_{\pi\in\Pi} E_{(s,a)\sim\mu^{\pi_t}}\left[\left(A^{\pi_t}(s,a) + \frac{1}{\eta}\right) \log \frac{p^\pi(a|s)}{p^{\pi_t}(a|s)}\right]$.

The full derivation of this computation can be found in Proposition 4 of Appendix D. Unlike for the direct representation, the above expression only involves the logarithm of the importance sampling ratio $p^\pi(a|s) / p^{\pi_t}(a|s)$. This will result in more stable off-policy updates in Section 5. In the tabular, finite state-action case, the update is equal to $p^{\pi_{t+1}}(a|s) \propto p^{\pi_t}(a|s) \max(1 + \eta A^{\pi_t}(s,a), 0)$. Next, we handle arbitrary policy parameterizations and discuss the choice of η in Section 6.

5 Policy parameterization

For simple parameterizations such as tabular, the set Π is convex and the minimization in Eq. (3) can be done exactly. When using more complex policy parameterizations (e.g.

Algorithm 1 FMA-PG: Framework for policy optimization

```
Input: $\pi_0$, $T$, $m$ (inner-loops), $\eta$,$\alpha$
for $t \leftarrow 0$ to $T-1$ do
    Form the surrogate function $\ell_t^{\pi,\phi,\eta}(\theta)$
    Initialize inner-loop: $\omega_0 = \theta_t$
    for $k \leftarrow 0$ to $m$ do
        $\omega_{k+1} = \omega_k + \alpha \nabla_\omega \ell_t^{\pi,\phi,\eta}(\omega_k)$
    end
    $\theta_{t+1} = \omega_m \quad ; \quad \pi_{t+1} = \pi(\theta_{t+1})$
end
Return $\theta_T$
```

deep neural network), the set of realizable policies Π can become arbitrarily complicated and non-convex, making the projection in Eq. (3) infeasible. Instead, to handle arbitrary policy parameterizations, we assume that Π consists of policies that are realizable by a model parameterized by $\theta \in \mathbb{R}^d$. We will continue to use π to refer to a policy's functional representation whereas $\pi(\theta)$ will refer to the parametric realization of π. Note that any generic model (e.g. neural network) can be used to parameterize π and is implicit in the $\pi(\theta)$ notation. For the special case of the tabular parameterization, $\pi = \pi(\theta) = \theta$. The following two problems are equivalent: $\max_{\pi \in \Pi} J(\pi) = \max_{\theta \in \mathbb{R}^d} J(\pi(\theta))$.

With this reparameterization, no projection is required and the update in Eq. (3) can be written as a parametric, unconstrained optimization problem, with $\pi_t = \pi(\theta_t)$, $\pi_{t+1} = \pi(\theta_{t+1})$ and $\theta_{t+1} = \arg\max_{\theta \in \mathbb{R}^d} \ell_t^{\pi,\phi,\eta}(\theta)$ where $\ell_t^{\pi,\phi,\eta}(\theta)$ depends on the functional representation, the mirror map, and η and is defined as $\ell_t^{\pi,\phi,\eta}(\theta) := J(\pi(\theta_t)) + \langle \pi(\theta) - \pi(\theta_t), \nabla_\pi J(\pi_t)\rangle - \frac{1}{\eta} D_\phi(\pi(\theta), \pi_t)$.

Compared to Eq. (3), we added terms independent of θ which do not change the $\arg\max$ but are useful to derive guarantees. The above objective is non-concave in general and can be maximized using a gradient-based algorithm. We will use m gradient steps with a step-size α to maximize $\ell_t^{\pi,\phi,\eta}(\theta)$. The overall algorithm is referred to as FMA-PG and its pseudo-code is given in Algorithm 1. Observe that the policy's functional representation affects the gradient $\nabla_\pi J(\pi_t)$. It is used to form the function $\ell_t^{\pi,\phi,\eta}$ that acts as a "guide" for the parametric updates in the inner-loop, similar to the recent algorithm proposed for supervised learning [13]. FMA-PG can be used with any functional representation or policy parameterization. We now consider the special case of the direct and softmax function representations and derive the corresponding $\ell_t^{\pi,\phi,\eta}$ functions.

Specifically, we parameterize the direct functional representation as $p^\pi(\cdot|s,\theta)$. Noting that $p^{\pi_t}(\cdot|s) = p^\pi(\cdot|s,\theta_t)$, $\ell_t^{\pi,\phi,\eta}(\theta) = \mathbb{E}_{(s,a)\sim\mu^{\pi_t}} \Big[\Big(Q^{\pi_t}(s,a) \frac{p^\pi(a|s,\theta)}{p^\pi(a|s,\theta_t)} \Big) - \frac{1}{\eta} D_\phi(p^\pi(\cdot|s,\theta), p^\pi(\cdot|s,\theta_t)) \Big] + C$, with C a constant independent of θ.

The above equation shows the benefits of first casting policy gradient methods as functional mirror ascent, and then using parametric updates to solve the resulting projection sub-problem. If we had directly chosen a parameterization and optimized $J(\pi(\theta))$ over θ, each parametric update would require computing the gradient and consequently involve collecting samples from the current policy $\pi(\theta_t)$. This makes the PG methods in [2; 21] computationally expensive. These methods can be obtained by FMA-PG with $m = 1$. For multiple steps, $m > 1$, FMA-PG requires parametric updates for $\ell_t(\theta)$. These updates rely on the states sampled from the fixed policy π_t. This natural *off-policyness* is an important feature of commonly used PG methods such as TRPO [25], PPO [26] and enable policy updates without interacting with the environment.

With a direct functional representation and the negative entropy as the mirror map, FMA-PG is similar to the algorithm proposed in MDPO [32]. However, the above formulation and MDPO have two main shortcomings. First, it involves $p^\pi(a|s,\theta)$ which means that for each parametric update, either (i) the actions need to be resampled on-policy or (ii) the update involves an importance-sampling ratio $p^\pi(a|s,\theta)/p^\pi(a|s,\theta_t)$. This requires clipping the ratio for stability, and can result in potentially conservative updates [26]. With the mirror map as the negative entropy, the Bregman divergence is the *reverse* KL divergence, i.e. $D_\phi(p^\pi(\cdot|s,\theta), p^\pi(\cdot|s,\theta_t)) = \text{KL}(p^\pi(\cdot|s,\theta)||p^\pi(\cdot|s,\theta_t))$. The reverse KL divergence makes this objective *mode seeking*, in that the policy π might only capture a subset of the actions covered by π_t. Past works have addressed this issue either by adding entropy regularization [10; 27], or by simply reversing the KL, using the *forward KL* - $\text{KL}(p^\pi(\cdot|s,\theta_t)||p^\pi(\cdot|s,\theta))$ [21]. However, using entropy regularization will result in a biased policy, whereas forward KL does not correspond to a valid Bregman divergence in p^π and can converge to a sub-optimal policy. We now show how FMA-PG with the softmax representation addresses both these issues in a principled way.

For the softmax functional representation considered in Section 4.2, we parameterize the logits as $z^\pi(a,s,\theta)$. Defining $p^\pi(a|s,\theta) = z^\pi(a,s,\theta)/\sum_{a'} z^\pi(a',s,\theta)$ and noting that $p^{\pi_t}(a|s) = p^\pi(a|s,\theta_t)$,

$$\ell_t^{\pi,\phi,\eta}(\theta) = \mathbb{E}_{s\sim d^{\pi_t}} \Bigg[\mathbb{E}_{a\sim p^{\pi_t}} \left(A^{\pi_t}(s,a) \log \frac{p^\pi(a|s,\theta)}{p^\pi(a|s,\theta_t)} \right) - \frac{1}{\eta} \text{KL}(p^\pi(\cdot|s,\theta_t)||p^\pi(\cdot|s,\theta)) \Bigg] + C.$$

Unlike for the direct formulation, we see that the above expression relies on the logarithm of the importance sampling ratios. Moreover, we observe that the KL divergence is in the *forward* direction and is *mode covering*. This naturally prevents a mode-collapse of the policy π and encourages exploration. We thus see that FMA-PG recovers the forward vs reverse KL heuristic in a principled manner.

Compared to TRPO [25], we observe that the above expression involves the logarithm of p^π. When the policy is modeled by a deep network with a final softmax layer, this leads to an objective concave in the last layer, which is easier to optimize than the TRPO objective. Unlike TRPO, the proposed update enforces the proximity between policies via a regularization rather than a constraint, a modification found to be beneficial [19]. Instead of tuning the hyper-parameters like for TRPO, the regularization strength $1/\eta$ in proposed update can be determined theoretically (Section 6).

6 Theoretical guarantees

In this section, we will discuss how to set η according to the properties of $J(\pi)$ for the direct and softmax functional representations, thus completely specifying $\ell_t^{\pi,\phi,\eta}(\theta)$.

FMA-PG obtains $\pi_{t+1} = \pi(\theta_{t+1})$ through the (potentially approximate) maximization of $\ell_t^{\pi,\phi,\eta}$. Observe that $J(\pi_t) = J(\pi(\theta_t)) = \ell_t^{\pi,\phi,\eta}(\theta_t)$. A sufficient condition to guarantee that $J(\pi_{t+1}) \geq J(\pi_t)$ is to make sure that $\ell_t^{\pi,\phi,\eta}(\theta) \leq J(\pi(\theta))$ for all θ. Indeed, if ℓ_t lower bounds J, improving ℓ_t leads to $J(\pi_{t+1}) = J(\pi(\theta_{t+1})) \geq \ell_t^{\pi,\phi,\eta}(\theta_{t+1}) \geq \ell_t^{\pi,\phi,\eta}(\theta_t) = J(\pi(\theta_t)) = J(\pi_t)$.

The following proposition shows that the values of η guaranteeing improvement are only dependent on properties of J and the mirror map in the function space and independent of the parameterization.

Proposition 2 (Smoothness and improvement guarantees)**.** *The surrogate function $\ell_t^{\pi,\phi,\eta}$ is a lower bound of J if and only if $J + \frac{1}{\eta}\phi$ is a convex function of π.*

The consequence of this proposition is that maximizing, even partially, $\ell_t^{\pi,\phi,\eta}(\theta)$ and hence $\ell_t^{\pi,\phi,\eta}(\theta)$ over θ, starting from $\theta = \theta_t$, guarantees $J(\pi_{t+1}) \geq J(\pi_t)$.

The following proposition details how the direct and softmax functional representations satisfy the conditions of Proposition 2 and thus offer improvement guarantees:

Proposition 3 (Improvement guarantees for direct and softmax representation)**.** *Assume that the rewards are in $[0, 1]$. Then both the direct and the softmax representation accept values of η that guarantee improvement:*

- Direct*: $J \geq \ell_t^{p^\pi,\phi,\eta}$ with ϕ the negative entropy and $\eta \leq \frac{(1-\gamma)^3}{2\gamma|A|}$.*
- Softmax*: $J \geq \ell_t^{z^\pi,\phi_z,\eta_z}$ with ϕ_z the exponential mirror map and $\eta_z \leq 1-\gamma$.*

Propositions 2 and 3 state that for specific values of η, any improvement to ℓ_t results in an improvement on the original objective. Moreover, these step-sizes only depend on the functional representation and the mirror map, and not on the particular parameterization chosen.

In order to show guaranteed improvement of $\pi_{t+1} = \pi(\theta_{t+1})$, we need to ensure that the parametric step-size α is chosen according to the smoothness of ℓ_t. With this, we obtain the following theorem:

Theorem 1 (Guaranteed improvement for parametric update)**.** *Assume that ℓ_t is β-smooth w.r.t. the Euclidean norm and that η satisfies the condition of Proposition 2. Then, for any $\alpha \leq 1/\beta$, iteration t of Algorithm 1 guarantees $J(\pi_{t+1}) \geq J(\pi_t)$ for any number m of inner loop updates.*

We see that even with an arbitrary parameterization, reparameterizing Eq. (3) into an unconstrained problem and solving it approximately with the correct choices of η and α guarantees improvement in $J(\pi)$ and results in convergence to a stationary point. Hence, a successful PG method relies on two different notions of smoothness, one at the policy level (to set η) and one at the parameter level (to set α). Note that Algorithm 1 and the corresponding theorem can be easily extended to handle stochastic parametric updates. This will guarantee that $\mathbb{E}[J(\pi_{t+1})] \geq J(\pi_t)$ where the expectation is over the sampling in the parametric SGD steps. Similarly, both the algorithm and theoretical guarantee can be generalized to incorporate the relative smoothness of $\ell_t(\theta)$ w.r.t. a general Bregman divergence.

Although we have used the same η for all states s, the updates can accommodate a different step-size $\eta(s)$ for each state. This is likely to yield tighter lower bounds and larger improvements in the inner loop. Determining such step-sizes is left for future work.

7 Conclusion

In this paper, we proposed FMA-PG, a general framework to design computationally efficient policy gradient methods. By disentangling the functional representation of a policy from its parameterization, we unified different PG perspectives, recovering several existing algorithms and implementation heuristics in a principled manner. We also enabled the design of new, improved PG methods, as testified by the strong results of the softmax formulation in various settings. By using the appropriate theoretically-determined hyper-parameters, FMA-PG guarantees policy improvement for the resulting PG method, even with arbitrarily complex policy parameterizations and for arbitrary number of inner loop steps. We believe this to be of great interest as it allows the natural design of sample-efficient, off-policy methods. Our theoretical results assume the exact computation of the action-value and advantage functions, and are thus, limiting in practice. In the future, we aim to handle sampling errors and extend these results to the actor-critic framework.

8 Acknowledgements

Nicolas Le Roux is supported by the Canada CIFAR AI Chair Program.

References

[1] Abdolmaleki, A., Springenberg, J. T., Tassa, Y., Munos, R., Heess, N., and Riedmiller, M. A. Maximum a posteriori policy optimisation. In *International Conference on Learning Representations (ICLR)*, 2018.

[2] Agarwal, A., Kakade, S. M., Lee, J. D., and Mahajan, G. Optimality and approximation with policy gradient methods in Markov decision processes. In *Conference on Learning Theory (COLT)*, pp. 64–66, 2020.

[3] Andrychowicz, M., Raichuk, A., Stanczyk, P., Orsini, M., Girgin, S., Marinier, R., Hussenot, L., Geist, M., Pietquin, O., Michalski, M., et al. What matters for on-policy deep actor-critic methods? a largescale study. In *International conference on learning representations*, 2021.

[4] Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[5] Bubeck, S. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

[6] Cen, S., Cheng, C., Chen, Y., Wei, Y., and Chi, Y. Fast global convergence of natural policy gradient methods with entropy regularization. *arXiv preprint arXiv:2007.06558*, 2020.

[7] Dhariwal, P., Hesse, C., Klimov, O., Nichol, A., Plappert, M., Radford, A., Schulman, J., Sidor, S., Wu, Y., and Zhokhov, P. Openai baselines. https://github.com/openai/baselines, 2017.

[8] Engstrom, L., Ilyas, A., Santurkar, S., Tsipras, D., Janoos, F., Rudolph, L., and Madry, A. Implementation matters in deep RL: A case study on PPO and TRPO. In *International conference on learning representations*, 2019.

[9] François-Lavet, V., Henderson, P., Islam, R., Bellemare, M. G., Pineau, J., et al. An introduction to deep reinforcement learning. *Foundations and Trends® in Machine Learning*, 11(3-4):219–354, 2018.

[10] Geist, M., Scherrer, B., and Pietquin, O. A theory of regularized Markov decision processes. In *International Conference on Machine Learning*, pp. 2160–2169. PMLR, 2019.

[11] Ghosh, D., C Machado, M., and Le Roux, N. An operator view of policy gradient methods. *Advances in Neural Information Processing Systems*, 33, 2020.

[12] Heess, N., Wayne, G., Silver, D., Lillicrap, T., Erez, T., and Tassa, Y. Learning continuous control policies by stochastic value gradients. In *Advances in Neural Information Processing Systems*, pp. 2944–2952, 2015.

[13] Johnson, R. and Zhang, T. Guided learning of nonconvex models through successive functional gradient optimization. In *International Conference on Machine Learning*, pp. 4921–4930. PMLR, 2020.

[14] Kakade, S. A natural policy gradient. In *NIPS*, volume 14, pp. 1531–1538, 2001.

[15] Kakade, S. and Langford, J. Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning (ICML)*, pp. 267–274, 2002.

[16] Kakade, S. M. A natural policy gradient. In *Advances in neural information processing systems*, pp. 1531–1538, 2002.

[17] Khodadadian, S., Jhunjhunwala, P. R., Varma, S. M., and Maguluri, S. T. On the linear convergence of natural policy gradient algorithm. *arXiv preprint arXiv:2105.01424*, 2021.

[18] Konda, V. R. and Tsitsiklis, J. N. Actor-critic algorithms. In *Advances in neural information processing systems*, pp. 1008–1014, 2000.

[19] Lazić, N., Hao, B., Abbasi-Yadkori, Y., Schuurmans, D., and Szepesvári, C. Optimization issues in kl-constrained approximate policy iteration. *arXiv preprint arXiv:2102.06234*, 2021.

[20] Lu, H., Freund, R. M., and Nesterov, Y. Relatively smooth convex optimization by first-order methods, and applications. *SIAM Journal on Optimization*, 28 (1):333–354, 2018.

[21] Mei, J., Xiao, C., Huang, R., Schuurmans, D., and Müller, M. On principled entropy exploration in policy optimization. In *IJCAI*, pp. 3130–3136, 2019.

[22] Mei, J., Xiao, C., Szepesvari, C., and Schuurmans, D. On the global convergence rates of softmax policy gradient methods. In *International Conference on Machine Learning*, pp. 6820–6829. PMLR, 2020.

[23] Neu, G., Jonsson, A., and Gómez, V. A unified view of entropy-regularized Markov decision processes. *CoRR*, abs/1705.07798, 2017.

[24] Puterman, M. L. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., USA, 1994.

[25] Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *International Conference on Machine Learning (ICML)*, pp. 1889–1897, 2015.

[26] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

[27] Shani, L., Efroni, Y., and Mannor, S. Adaptive trust region policy optimization: Global convergence and faster rates for regularized mdps. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5668–5675, 2020.

[28] Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. Deterministic policy gradient algorithms. *Journal of Machine Learning Research*, 2014.

[29] Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. MIT Press, 2 edition, 2018.

[30] Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1057–1063, 2000.

[31] Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012.

[32] Tomar, M., Shani, L., Efroni, Y., and Ghavamzadeh, M. Mirror descent policy optimization. *arXiv preprint arXiv:2005.09814*, 2020.

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

[34] Williams, R. J. and Peng, J. Function optimization using connectionist reinforcement learning algorithms. *Connection Science*, 3(3):241–268, 1991.

Supplementary material

Organization of the Appendix

A Experiments using FMA-PG with the softmax functional representation

While this work focuses on providing a general framework for designing PG methods, we explore the behaviour of FMA-PG with the softmax functional representation. For continuous-control tasks on the Mujoco environment, we modify the update to make it similar to PPO [26], calling the resulting algorithm *sPPO*.

For the sPPO update, the ℓ_t function in Algorithm 1 is given by: $\ell_t^{\pi,\phi,\eta}(\theta) = \mathbb{E}_{(s,a)\sim\mu^{\pi_t}}\left[A^{\pi_t}(s,a)\log\left(\text{clip}\left(\frac{p^\pi(a|s,\theta)}{p^\pi(a|s,\theta_t)}, \frac{1}{1+\epsilon}, 1+\epsilon\right)\right)\right]$ where the importance weight is clipped to the $[\frac{1}{1+\epsilon}, 1+\epsilon]$ range, like PPO. We investigate the performance of sPPO on five standard continuous control environments from the OpenAI Gym suite [4]: Hopper-v1, Walker2d-v1, HalfCheetah-v1, Ant-v1, and Humanoid-v1. As a baseline, we use the PPO implementation from Andrychowicz et al. [3] with their standard configuration and all the hyperparameters set to the default values in Table 2 of Appendix C of [3]. We implement sPPO by adding a binary flag (`use_softmax`).

We investigate the differences between PPO and sPPO by training 180 different policies for each environment and all combinations of `use_softmax` $\in \{$`True`, `False`$\}$, $m \in \{10, 100\}$ and the importance weight capping value $\epsilon \in \{0.1, 0.3, 0.5, 0.7\}$ (a total compute of 1400 days with TPUv2). We evaluate each policy 18 times during training, using the action with largest probability rather than a sample. We compute the average return and 95% confidence intervals for each of the settings. The results are presented in Fig. 1, where we see that sPPO outperforms PPO across all environments. Furthermore, we see that the difference is more pronounced when the number of iterations m in the inner loop is increased (linestyles) or when less capping is used (columns).

In Fig. 2, we show additional results but with learning rate decay and gradient clipping disabled, two commonly used techniques to stabilize PPO training [8]. In this setting, sPPO only suffers a mild degradation while PPO fails completely, again confirming the additional robustness of sPPO compared to PPO.

B Handling stochastic value gradients

Thus far we have worked with the original formulation of policy gradients where a policy is a distribution over actions given states. An alternative approach is that taken by stochastic value gradients [12], that rely on the reparametrization trick. In this case, a policy is not represented by a distribution over actions but rather by a set of actions. Formally, if ε are random variables drawn from a fixed distribution ν, then policy π is a deterministic map from $\mathcal{S} \times \nu \to \mathcal{A}$. This corresponds to the functional representation of the policy. The action a chosen by π in state s (when fixing the random variable $\epsilon = \varepsilon$) is represented as $\pi(s, \epsilon)$ and

$$J(\pi) = \sum_s d^\pi(s) \int_\varepsilon \nu(\varepsilon)\, r(s, \pi(s,\varepsilon))\, d\varepsilon \tag{4}$$

and Silver et al. [28] showed that $\dfrac{\partial J(\pi)}{\partial \pi(s,\epsilon)} = d^\pi(s)\nabla_a Q^\pi(s,a)\big|_{a=\pi(s,\epsilon)}$.

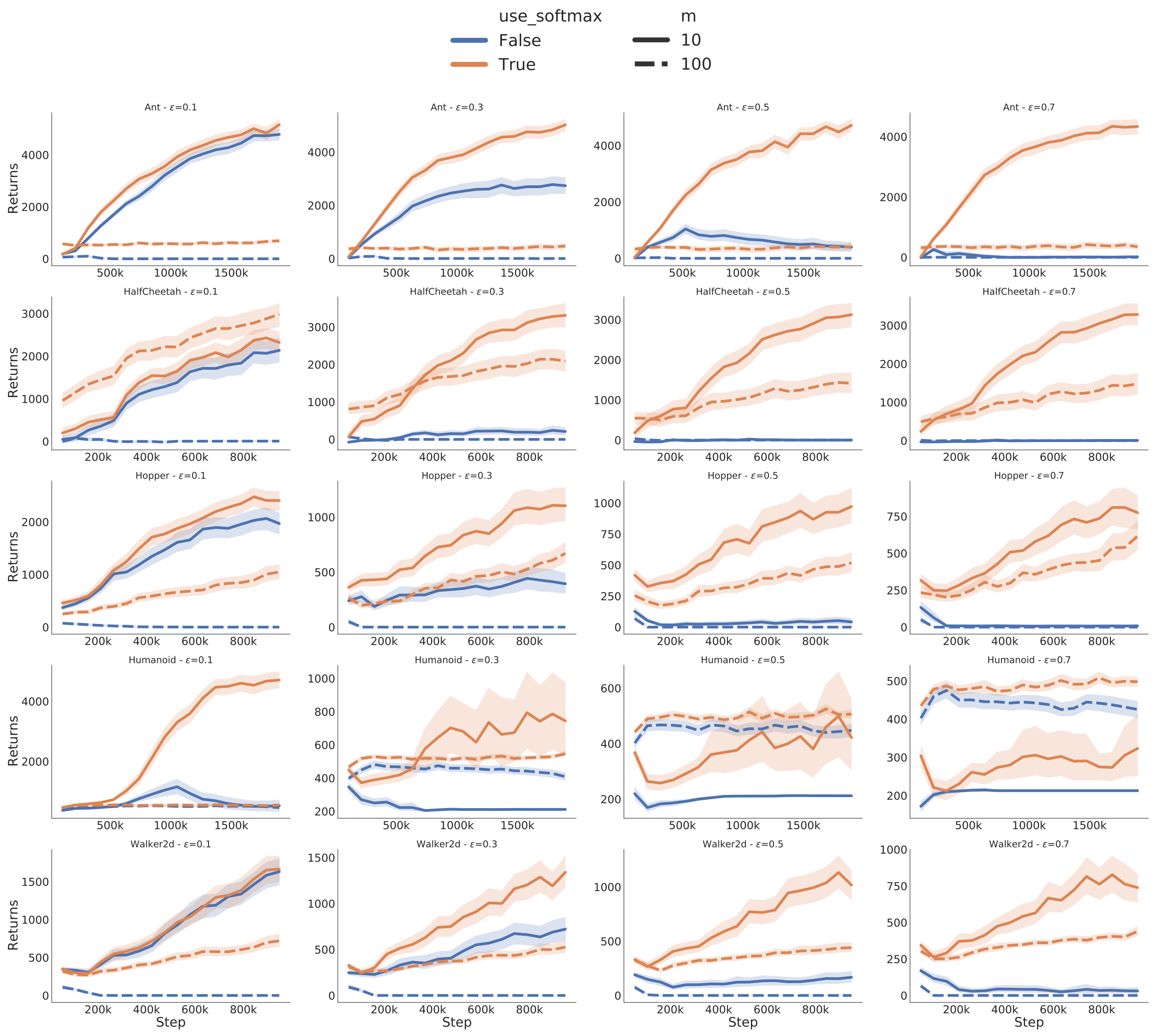

Figure 1: Average return and 95% confidence intervals (over 180 runs) for PPO and sPPO on 5 environments rows) and for four different clipping values (columns). sPPO is more robust to large values of clipping, even more so when the number of updates in the inner loop grows (linestyle).

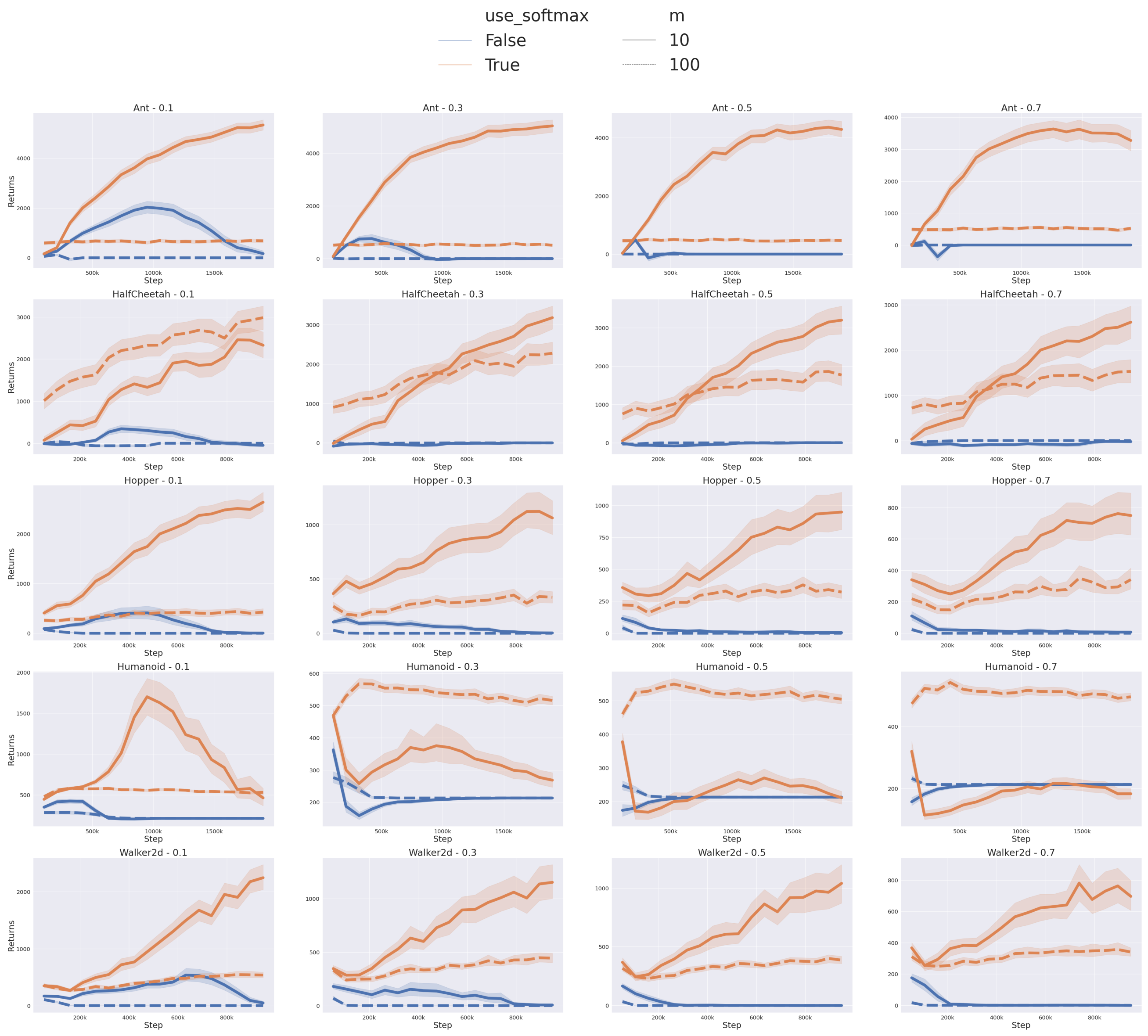

Figure 2: Average discounted return and 95% confidence interval (over 180 runs) for PPO and softmax PPO on 4 environments (*env* - rows) and for four different clipping strengths (*epsilon* - columns).

If the policy π is parameterized by model f with parameters θ, then $\pi(s, \epsilon) = f(\theta, s, \epsilon)$. If $f(\theta_t, \epsilon)$ and $f(\theta, \epsilon)$ are S-dimensional vectors, then Eq. (3) is given as

$$\theta_{t+1} = \arg\min \mathbb{E}_{\epsilon \sim \nu} \left[-\sum_s d^{\pi_t}(s) f(\theta, s, \epsilon) \nabla_a Q^{\pi_t}(s, a)\big|_{a = f(\theta_t, s, \epsilon)} + \frac{1}{\eta} D_\phi(f(\theta, \epsilon), f(\theta_t, \epsilon)) \right] . \tag{5}$$

Similar to Sections 4.1 and 4.2, we will use a mirror map that decomposes across states. Specifically, we choose $D_\phi(\pi, \mu) = \sum_{s \in \mathcal{S}} d^{\pi_t}(s) \, ||\pi(s) - \mu(s)||^2$. With this choice, Eq. (5) can be written as:

$$\theta_{t+1} = \arg\max \left[\mathbb{E}_{s \sim d^{\pi_t}} \left[\mathbb{E}_{\epsilon \sim \nu} \left[f(\theta, s, \epsilon) \nabla_a Q^{\pi_t}(s, a)\big|_{a = f(\theta_t, s, \epsilon)} - \frac{1}{\eta} \, ||f(\theta, \epsilon) - f(\theta_t, \epsilon)||^2 \right] \right] \right] \tag{6}$$

This formulation is similar to Eq (15) of [28], with Q^{π_t} instead of Q^π. Additionally, while the authors justified the off-policy approach with an approximation, our formulation offers guarantees provided η satisfies the condition of Proposition Proposition 2.

C Proofs for Section 3

Proposition 1 (Operator consistency for the FMA update). *By defining the improvement and projection operators as the update and projection step of FMA, for the same mirror map (as in Eqs. (1) and (2)), π^* is a fixed point of $\mathcal{P} \circ \mathcal{I}$.*

Proof. Since π^* is the optimal policy, it is a stationary point of $J(\pi)$, implying that $\nabla J(\pi^*) = 0$. If we use the FMA update in Eq. (1) with $\pi_t = \pi^*$, then,

$$\pi_{t+1/2} = (\nabla \Phi)^{-1} \left(\nabla \phi(\pi^*) + \eta \nabla J(\pi^*) \right) \implies \pi_{t+1/2} = \pi^*.$$

For the projection in Eq. (2), using the above relation,

$$\pi_{t+1} = \arg\min_{\pi \in \Pi} D_\phi(\pi, \pi^*) \implies \forall \pi \in \Pi, D_\phi(\pi_{t+1}, \pi^*) \leq D_\phi(\pi, \pi^*)$$

Since $\pi^* \in \Pi$, $\min_{\pi \in \Pi} D_\phi(\pi, \pi^*) = 0$, $\implies$ $D_\phi(\pi_{t+1}, \pi^*) \leq 0$. Since the Bregman divergence is non-negative, $D_\phi(\pi_{t+1}, \pi^*) \implies \pi_{t+1} = \pi^*$. The above relations imply that if $\pi_t = \pi^*$, the FMA update ensures that $\pi_{t+1} = \pi^*$ and hence π^* is a fixed point of $\mathcal{P} \circ \mathcal{I}$.

□

D Proofs for Section 4

In this section, we prove the equivalence of the formulations in terms of the logits and in terms of $\log \pi$.

Lemma 1. *Let*

$$\phi(z) = \frac{\sum_a \exp(z(a))}{\sum_a \exp(z'(a))} \tag{7}$$

$$p^\pi(a) = \frac{\exp(z(a))}{\sum_{a'} \exp(z(a'))} , \tag{8}$$

for some fixed z'. Then

$$D_\phi(z, z') = KL(p^{\pi'} || p^\pi) + \Delta \tag{9}$$

where p^π and $p^{\pi'}$ use z and z' respectively, z' is the one used in the denominator of the mirror map, and $\Delta \leq 0$ is independent of p^π.

Proof.

$$\begin{aligned}
D_\phi(z,z') &= \frac{\sum_a \exp(z(a))}{\sum_a \exp(z'(a))} - \frac{\sum_a \exp(z'(a))}{\sum_a \exp(z'(a))} - \frac{\sum_a \exp(z'(a))(z(a)-z'(a))}{\sum_a \exp(z'(a))}\\
&= \frac{\sum_a \exp(z(a))}{\sum_a \exp(z'(a))} - 1 - \sum_a p^{\pi'}(a)(z(a)-z'(a))\\
&= \frac{\sum_a \exp(z(a))}{\sum_a \exp(z'(a))} - \sum_a p^{\pi'}(a)(z(a)-\delta-z'(a)+\delta') - 1 - \sum_a p^{\pi'}(a)(\delta-\delta')\ ,
\end{aligned}$$

where the last equation is true for all δ and all δ'. By choosing

$$\begin{aligned}
\delta &= \log\left(\sum_a \exp(z(a))\right)\\
\delta' &= \log\left(\sum_a \exp(z'(a))\right)\ ,
\end{aligned}$$

we have

$$z(a) - \delta = \log p^\pi(a)\ ,$$

and

$$\begin{aligned}
D_\phi(z,z') &= \exp(\delta-\delta') - \sum_a p^{\pi'}(a)\log\frac{p^\pi(a)}{p^{\pi'}(a)} - 1 + \delta' - \delta\\
&= KL(p^{\pi'}||p^\pi) + \exp(\delta-\delta') - 1 + \delta' - \delta\ .
\end{aligned}$$

Shifting all values of z by the same amount affects δ but not p^π because of the normalization. Hence, $\exp(\delta-\delta')-1+\delta'-\delta$ is independent of p^π.

Finally, we use that $\exp(x)-1-x \geq 0$ for all x with $x=\delta-\delta'$ to conclude the proof. □

Proposition 4.

$$\ell_t^{z^\pi,\phi,\eta}(\theta) = J(\pi_t) + E_{(s,a)\sim\mu^{\pi_t}}\left(A^{\pi_t}(s,a) + \frac{1}{\eta}\right)\log\frac{p^\pi(a|s,\theta)}{p^{\pi_t}(a|s,\theta)} - \Delta\ , \tag{10}$$

with $\Delta \leq 0$ a constant independent of π.

Proof. Because $\sum_a p^{\pi_t}(a|s)A^{\pi_t}(s,a) = 0$, we can shift all values of z by a term that does not depend on a without changing the sum, in particular by $\log\left(\sum_{a'}\exp(z^\pi(a',s|\theta)\right)$. Thus,

$$\begin{aligned}
\ell_t^{z^\pi,\phi,\eta}(\theta) &= J(\pi_t) + E_{(s,a)\sim\mu^{\pi_t}}A^{\pi_t}(s,a)\left(z^\pi(a,s|\theta) - \log\left(\sum_{a'}\exp(z^\pi(a',s|\theta))\right)\right)\\
&\quad - \frac{1}{\eta}\sum_s d^{\pi_t}(s)D_{\phi_z}(z^\pi(\cdot,s|\theta), z^\pi(\cdot,s,\theta_t))\\
&= J(\pi_t) + E_{(s,a)\sim\mu^{\pi_t}}A^{\pi_t}(s,a)\log p^\pi(a|s,\theta) - \frac{1}{\eta}\sum_s d^{\pi_t}(s)D_{\phi_z}(z^\pi(\cdot,s|\theta), z^\pi(\cdot,s|\theta_t))\ .
\end{aligned}$$

Using Lemma 1, we have

$$D_{\phi_z}(z^\pi(\cdot,s|\theta), z^\pi(\cdot,s|\theta_t)) = \sum_s d^{\pi_t}(s)\left(KL(p^{\pi'}(\cdot|s)||p^\pi(\cdot|s)) + \exp(\delta(s)-\delta'(s)) - 1 + \delta' - \delta\right)\ ,$$

for some δ and δ' independent of p^π.

Noting that

$$KL(p^{\pi'}(\cdot|s)||p^\pi(\cdot|s) = \sum_a p^{\pi_t}(a|s) \log \frac{p^\pi(a|s)}{p^{\pi_t}(a|s)} ,$$

$$\ell_t^{z^\pi,\phi,\eta}(\theta) = J(\pi_t) + E_{(s,a)\sim\mu^{\pi_t}} \left(A^{\pi_t}(s,a) + \frac{1}{\eta}\right) \log \frac{p^\pi(a|s,\theta)}{p^{\pi_t}(a|s,\theta)} - \Delta ,$$

with $\Delta \leq 0$ independent of p^π. This concludes the proof. $\square$

E Proofs for Section 6

Proposition 2 (Smoothness and improvement guarantees). *The surrogate function $\ell_t^{\pi,\phi,\eta}$ is a lower bound of J if and only if $J + \frac{1}{\eta}\phi$ is a convex function of π.*

Proof.

$$\begin{aligned}
J(\pi) - \ell_t^{\pi,\phi,\eta}(\pi) &= J(\pi) - J(\pi_t) - \langle \pi - \pi_t, \nabla_\pi J(\pi_t)\rangle + \frac{1}{\eta} D_\phi(\pi, \pi_t) \\
&= J(\pi) - J(\pi_t) - \langle \pi - \pi_t, \nabla_\pi J(\pi_t)\rangle + \frac{1}{\eta}\left(\phi(\pi) - \phi(\pi_t) - \langle \nabla_\pi \phi(\pi_t), \pi - \pi_t\rangle\right) \\
&= \left(J + \frac{1}{\eta}\phi\right)(\pi) - \left(J + \frac{1}{\eta}\phi\right)(\pi_t) - \langle \pi - \pi_t, \nabla_\pi \left(J + \frac{1}{\eta}\phi\right)(\pi_t)\rangle .
\end{aligned}$$

The last equation is positive for all π and all π_t if and only if $J + \frac{1}{\eta}\phi$ is convex. $\square$

To prove the value of η guaranteeing improvement for the softmax parameterization, we first need to extend a lower bound result from Ghosh et al. [11]:

Proposition 5. *Let us assume that the rewards are lower bounded by $-c$ for some $c \in \mathbb{R}$. Then we have*

$$J(\pi) \geq J(\pi_t) + E_{(s,a)\sim\mu^{\pi_t}} \left[\left(Q^{\pi_t}(s,a) + \frac{c}{1-\gamma}\right) \log \frac{p^\pi(a|s)}{p^{\pi_t}(a|s)}\right] . \tag{11}$$

Proof. Let us define the function J_ν for a policy ν as

$$J_\nu(\pi) = \sum_{h=0}^{+\infty} \gamma^h \int_{\tau_h} (r(s_h, a_h) + c) \left(1 + \log \frac{\pi_h(\tau_h)}{\nu_h(\tau_h)}\right) \nu_h(\tau_h)\, d\tau_h - \frac{c}{1-\gamma} ,$$

where τ_h is a trajectory of length h that is a prefix of a full trajectory τ and π_h is the policy restricted to trajectories of length h. We first show that it satisfies $J_\nu(\pi) \leq J(\pi)$ for any ν and any π such that the support of ν covers that of π.

Indeed, we can rewrite

$$\begin{aligned}
J(\pi) &= \int_\tau \left(R(\tau) + \frac{c}{1-\gamma}\right) \pi(\tau)\, d\tau - \frac{c}{1-\gamma} \\
&= \int_\tau \left(\sum_h \gamma^h (r(a_h, s_h) + c)\right) \pi(\tau)\, d\tau - \frac{c}{1-\gamma} && \text{(using } \textstyle\sum_h \gamma^h c = c/(1-\gamma)) \\
&= \sum_h \gamma^h \int_{\tau_h} (r(a_h, s_h) + c)\pi_h(\tau_h)\, d\tau_h - \frac{c}{1-\gamma} ,
\end{aligned}$$

where the last line is obtained by marginalizing over steps $h+1,\dots,+\infty$ for all h and all trajectories τ. Because $r(a_h, s_h)+c$ is positive, as the rewards are lower bounded by $-c$, we have

$$\begin{aligned} J(\pi) &= \sum_h \gamma^h \int_{\tau_h} (r(a_h, s_h)+c)\frac{\pi_h(\tau_h)}{\nu_h(\tau_h)}\nu_h(\tau_h)\, d\tau_h - \frac{c}{1-\gamma} \\ &\geq \sum_h \gamma^h \int_{\tau_h} (r(a_h, s_h)+c)\left(1+\log\frac{\pi_h(\tau_h)}{\nu_h(\tau_h)}\right)\nu_h(\tau_h)\, d\tau_h - \frac{c}{1-\gamma} && \text{(using } x \geq 1+\log x\text{)} \\ &= J_\nu(\pi)\ . \end{aligned}$$

Let us denote J_ν^{SA} the right-hand side of Eq. (11), i.e.:

$$J_\nu^{SA}(\pi) = J(\nu) + E_{(s,a)\sim\mu^\nu}\left[\left(Q^\nu(s,a)+\frac{c}{1-\gamma}\right)\log\frac{p^\pi(a|s)}{p^\nu(a|s)}\right]\ .$$

We now prove that J_ν has the same gradient as J_ν^{SA}:

$$\begin{aligned} \nabla_\theta J_\nu(\pi) &= \nabla_\theta\left(\sum_h \gamma^h \int_{\tau_h} (r(a_h, s_h)+c)\left(1+\log\frac{\pi_h(\tau_h)}{\nu_h(\tau_h)}\right)\nu_h(\tau_h)\, d\tau_h\right) \\ &= \nabla_\theta\left(\sum_h \gamma^h \int_{\tau_h} (r(a_h, s_h)+c)\log\pi_h(\tau_h)\nu_h(\tau_h)\right)\, d\tau_h \\ &= \sum_h \gamma^h \int_{\tau_h} (r(a_h, s_h)+c)\nabla_\theta\log\pi_h(\tau_h)\nu_h(\tau_h)\, d\tau_h\ , \end{aligned}$$

where all terms independent of θ were moved outside of the gradient. As the log probability of a trajectory decomposes into a sum of the probabilities of actions given states and of the transition probabilities, and as the latter are independent of θ, we get

$$\begin{aligned} \nabla_\theta J_\nu(\pi) &= \sum_h \gamma^h \int_{\tau_h} (r(a_h, s_h)+c)\nabla_\theta\log\pi_h(\tau_h)\nu_h(\tau_h)\, d\tau_h \\ &= \sum_h \gamma^h \int_{\tau_h} (r(a_h, s_h)+c)\left(\sum_{h'}\nabla_\theta\log p^\pi(a_{h'}|s_{h'})\right)\nu_h(\tau_h)\, d\tau_h \\ &= \int_\tau \sum_{h'}\nabla_\theta\log p^\pi(a_{h'}|s_{h'})\left(\sum_{h=h'}^{+\infty}\gamma^h(r(a_h, s_h)+c)\right)\nu(\tau)\, d\tau\ . \end{aligned}$$

But

$$\begin{aligned} &\sum_{h=h'}^{+\infty}\gamma^h(r(a_h, s_h)+c) = \gamma^{h'}\left(Q^\nu(s,a)+\frac{c}{1-\gamma}\right) \\ &\int_\tau \nu(\tau)d\tau 1_{a_{h'}=a}1_{s_{h'}=s} = d_\nu^{h'}(s)\nu(a|s)\ , \end{aligned}$$

with $d_\nu^{h'}(s)$ the undiscounted probability of reaching state s at timestep h'. Hence, we have

$$\begin{aligned}
\nabla_\theta J_\nu(\pi) &= \int_\tau \sum_{h'} \nabla_\theta \log p^\pi(a_{h'}|s_{h'}) \left(\sum_{h=h'}^{+\infty} \gamma^h (r(a_h, s_h) + c)\right) \nu(\tau)\, d\tau \\
&= \sum_{h'} \sum_s \sum_a \nabla_\theta \log p^\pi(a|s) d_\nu^{h'}(s) \nu(a|s) \gamma^{h'} \left(Q^\nu(s,a) + \frac{c}{1-\gamma}\right) \\
&= \sum_{h'} \gamma^{h'} \sum_s d_\nu^{h'}(s) \sum_a \nabla_\theta \log p^\pi(a|s) \nu(a|s) \left(Q^\nu(s,a) + \frac{c}{1-\gamma}\right) \\
&= \sum_s d^\nu(s) \sum_a \left(Q^\nu(s,a) + \frac{c}{1-\gamma}\right) \nu(a|s) \nabla_\theta \log p^\pi(a|s) \\
&= \nabla_\theta \left(\sum_s d^\nu(s) \sum_a \left(Q^\nu(s,a) + \frac{c}{1-\gamma}\right) \nu(a|s) \log p^\pi(a|s)\right) \\
&= \nabla_\theta \left(J(\nu) + E_{(s,a)\sim\mu^\nu}\left[\left(Q^\nu(s,a) + \frac{c}{1-\gamma}\right) \log \frac{p^\pi(a|s)}{p^\nu(a|s)}\right]\right) \\
&= \nabla_\theta J_\nu^{SA}(\pi)\ ,
\end{aligned}$$

with $d^\nu(s)$ the unnormalized probability of s under the *discounted* stationary distribution.

Because J_ν and J_ν^{SA} have the same gradient, they differ by a constant, i.e. $J_\nu^{SA} = J_\nu + C$ for some C. But we also know that $J_\nu(\nu) = J(\nu)$, which means that

$$\begin{aligned}
C &= J_\nu^{SA}(\nu) - J_\nu(\nu) \\
&= J_\nu^{SA}(\nu) - J(\nu) \\
&= E_{(s,a)\sim\mu^\nu}\left[\left(Q^\nu(s,a) + \frac{c}{1-\gamma}\right) \log \frac{p^\nu(a|s)}{p^\nu(a|s)}\right] \\
&= 0\ .
\end{aligned}$$

Hence, $J_\nu = J_\nu^{SA}$ and, becomes J_ν is a lower bound of J, we have

$$J(\pi) \geq J(\nu) + \sum_s d^\nu(s) \sum_a \left(Q^\nu(s,a) + \frac{c}{1-\gamma}\right) p^\nu(a|s) \log \frac{p^\pi(a|s)}{p^\nu(a|s)}\ . \tag{12}$$

Setting $\nu = \pi_t$ concludes the proof. □

Proposition 3 (Improvement guarantees for direct and softmax representation)**.** *Assume that the rewards are in* $[0, 1]$*. Then both the direct and the softmax representation accept values of* η *that guarantee improvement:*

- Direct*:* $J \geq \ell_t^{p^\pi,\phi,\eta}$ *with* ϕ *the negative entropy and* $\eta \leq \frac{(1-\gamma)^3}{2\gamma|A|}$.
- Softmax*:* $J \geq \ell_t^{z^\pi,\phi_z,\eta_z}$ *with* ϕ_z *the exponential mirror map and* $\eta_z \leq 1-\gamma$.

Proof. Agarwal et al. [2] show that, when using the direct parameterization, J is $\left(\frac{2\gamma|A|}{(1-\gamma)^3}\right)$-smooth w.r.t. the Euclidean distance. By using the properties of relative smoothness [20], if the mirror map ϕ is μ-strongly convex w.r.t. Euclidean distance, then J is L-smooth with $L = \left({}^{2\gamma|A|}/{}_{(1-\gamma)^3}\, \mu\right)$. Using the fact that negative entropy is 1-strongly convex w.r.t. the 1-norm, we can set $\eta = {}^{(1-\gamma)^3}/{}_{2\gamma|A|}$ for the direct formulation.

We now prove the result for the softmax formulation. Assume

$$\eta = \frac{1-\gamma}{r_m - r_l}\ . \tag{13}$$

We know from Proposition 4 that

$$\ell_t^{z^\pi,\phi,\eta}(\theta) \le J(\pi_t) + E_{(s,a)\sim\mu^{\pi_t}} \left(A^{\pi_t}(s,a) + \frac{1}{\eta}\right) \log \frac{p^\pi(a|s,\theta)}{p^{\pi_t}(a|s,\theta)} \ .$$

Since the rewards are between r_l and r_m, we have

$$\begin{aligned}
\ell_t^{z^\pi,\phi,\eta}(\pi) &\le J(\pi_t) + E_{(s,a)\sim\mu^{\pi_t}} \left[\left(A^{\pi_t}(s,a) + \frac{1}{\eta}\right) \log \frac{p^\pi(a|s)}{p^{\pi_t}(a|s)}\right] \\
&= J(\pi_t) + E_{(s,a)\sim\mu^{\pi_t}} \left[\left(A^{\pi_t}(s,a) + \frac{r_m - r_l}{1-\gamma}\right) \log \frac{p^\pi(a|s)}{p^{\pi_t}(a|s)}\right] \\
&= J(\pi_t) + E_{(s,a)\sim\mu^{\pi_t}} \left[\left(A^{\pi_t}(s,a) + V^{\pi_t}(s) + \left(\frac{r_m}{1-\gamma} - V^{\pi_t}(s)\right) - \frac{r_l}{1-\gamma}\right) \log \frac{p^\pi(a|s)}{p^{\pi_t}(a|s)}\right] \\
&= J(\pi_t) + E_{(s,a)\sim\mu^{\pi_t}} \left[\left(Q^{\pi_t}(s,a) - \frac{r_l}{1-\gamma}\right) \log \frac{p^\pi(a|s)}{p^{\pi_t}(a|s)}\right] \\
&\quad - E_{s\sim d^{\pi_t}} \left[\left(\frac{r_m}{1-\gamma} - V^{\pi_t}(s)\right) KL(p^{\pi_t}(\cdot|s)||p^\pi(\cdot|s))\right] \ .
\end{aligned}$$

The last term on the RHS of the last equation is negative. Indeed, because the rewards are less than r_m, the value functions are less than $r_m/(1-\gamma)$ and $r_m/(1-\gamma) - V^{\pi_t}(s)$ is positive. As the KL divergences are positive, the product of the two is positive and the whole term is negative because of the minus term. Thus, we have

$$\begin{aligned}
\ell_t^{z^\pi,\phi,\eta}(\pi) &\le J(\pi_t) + E_{(s,a)\sim\mu^{\pi_t}} \left[\left(Q^{\pi_t}(s,a) - \frac{r_l}{1-\gamma}\right) \log \frac{p^\pi(a|s)}{p^{\pi_t}(a|s)}\right] \\
&\le J(\pi) \ . && \text{(by Proposition 5)}
\end{aligned}$$

Hence, choosing $\eta = \frac{1-\gamma}{r_m - r_l}$ leads to an improvement guarantee. Because our rewards are bounded between 0 and 1, setting $r_m = 1$ and $r_l = 0$ gives $\eta = 1-\gamma$. This concludes the proof. □

Theorem 1 (Guaranteed improvement for parametric update)**.** *Assume that ℓ_t is β-smooth w.r.t. the Euclidean norm and that η satisfies the condition of Proposition 2. Then, for any $\alpha \le 1/\beta$, iteration t of Algorithm 1 guarantees $J(\pi_{t+1}) \ge J(\pi_t)$ for any number m of inner loop updates.*

Proof. Using the update in Algorithm 1 with $\alpha \le \frac{1}{\beta}$ and the β-smoothness of $\ell_t(\omega)$, for all $k \in [m-1]$,

$$\ell_t(\omega_{k+1}) \ge \ell_t(\omega_k) + \frac{1}{2\beta}||\nabla \ell_t(\omega_k)||^2$$

After m steps,

$$\ell_t(\omega_m) \ge \ell_t(\omega_0) + \frac{1}{2\beta}\sum_{k=0}^{m-1}||\nabla \ell_t(\omega_k)||^2$$

Since $\theta_{t+1} = \omega_m$ and $\omega_0 = \theta_t$ in Algorithm 1,

$$\implies \ell_t(\theta_{t+1}) \ge \ell_t(\theta_t) + \frac{1}{2\beta}||\nabla \ell_t(\theta_t)||^2 + \sum_{k=1}^{m-1}||\nabla \ell_t(\omega_k)||^2$$

Note that $J(\pi_t) = \ell_t(\theta_t)$ and if η satisfies Proposition 2, then $J(\pi_{t+1}) \ge \ell_t(\theta_{t+1})$. Using these relations,

$$J(\pi_{t+1}) \ge J(\pi_t) + \underbrace{\frac{1}{2\beta}||\nabla \ell_t(\theta_t)||^2 + \sum_{k=1}^{m-1}||\nabla \ell_t(\omega_k)||^2}_{\text{+ve}} \implies J(\pi_{t+1}) \ge J(\pi_t).$$

□