
Fast Convergence of Softmax Policy Mirror Ascent

Reza Asad¹ Reza Babanezhad² Issam Laradji³ Nicolas Le Roux⁴ Sharan Vaswani¹
¹Simon Fraser University ² Samsung AI ³ ServiceNow ⁴Mila, Université de Montréal, McGill

Abstract

Natural policy gradient (NPG) is a common policy optimization algorithm and can be viewed as mirror ascent in the space of probabilities. Recently, Vaswani et al. (2021) introduced a policy gradient method that corresponds to mirror ascent in the dual space of logits. We refine this algorithm, removing its need for a normalization across actions and analyze the resulting method (referred to as SPMA). For tabular MDPs, we prove that SPMA with a constant step-size matches the linear convergence of NPG and achieves a faster convergence than constant step-size (accelerated) softmax policy gradient. To handle large state-action spaces, we extend SPMA to use a log-linear policy parameterization. Unlike that for NPG, generalizing SPMA to the linear function approximation (FA) setting does not require compatible function approximation. Unlike MDPO, a practical generalization of NPG, SPMA with linear FA only requires solving convex softmax classification problems. We prove that SPMA achieves linear convergence to the neighbourhood of the optimal value function. We extend SPMA to handle non-linear FA and evaluate its empirical performance on the MuJoCo and Atari benchmarks. Our results demonstrate that SPMA consistently achieves similar or better performance compared to MDPO, PPO and TRPO.

1 INTRODUCTION

Policy gradient (PG) methods (Williams, 1992; Sutton et al., 1999; Konda and Tsitsiklis, 2000; Kakade, 2001) have been critical to the achievements of reinforcement learning (RL). Although the PG objective

is non-concave, recent theoretical research (Agarwal et al., 2021; Mei et al., 2020, 2021b; Bhandari and Russo, 2021; Lan, 2023; Shani et al., 2020; Liu et al., 2024; Lu et al., 2024; Alfano and Rebeschini, 2022; Yuan et al., 2023) has analyzed PG methods in simplified settings and demonstrated their global convergence to an optimal policy. While such simplified analyses are helpful in understanding the underlying optimization issues, the resulting methods are rarely used in practice. On the other hand, while methods such as TRPO (Schulman, 2015), PPO (Schulman et al., 2017), MDPO (Tomar et al., 2020) are commonly used in deep RL, their theoretical analysis in the function approximation setting is quite limited. In particular, existing work either (i) analyzes these methods only in the impractical tabular setting (Tomar et al., 2020; Shani et al., 2020) or (ii) modifies these algorithms to make them more amenable to theoretical analysis (Liu et al., 1906; Zhong and Zhang, 2024). Unfortunately, these modified algorithms are quite different from the original variants and are not systematically benchmarked on standard environments. Consequently, there exists a large gap between PG methods that have theoretical guarantees in realistic settings versus those which are implemented in practice. To make matters worse, it has been demonstrated that code-level implementation details impact the empirical performance more than the underlying algorithm (Engstrom et al., 2019).

Designing theoretically principled PG algorithms that simultaneously have good empirical performance on the standard set of benchmarks is the main motivation behind this work. To that end, we leverage an algorithm first proposed by Vaswani et al. (2021), which we modify to remove the need for normalization. We coin this refinement **Softmax Policy Mirror Ascent** (referred to as SPMA). We show that SPMA has comparable convergence guarantees as existing theoretical techniques (Lu et al., 2024; Yuan et al., 2023) in the tabular and function approximation settings, while achieving comparable practical performance as PPO, TRPO and MDPO, without additional algorithmic modifications. In particular, we make the following contributions.

Contribution 1: In Section 3, we focus on the multi-

armed bandit and tabular MDP settings, where the number of parameters scales with the number of states and actions. We develop the **SPMA** algorithm, which parameterizes the policy using the softmax function and uses a mirror ascent (with the log-sum-exp mirror map) update. Compared to **NPG** that can be viewed as mirror ascent in the space of probabilities, **SPMA** corresponds to mirror ascent in the dual space of logits and does not require a normalization across actions. Given access to the exact policy gradients, we prove that **SPMA** with a constant step-size converges to the optimal policy at a linear rate and thus matches the rate of **NPG** (Khodadadian et al., 2021; Liu et al., 2024). In comparison, constant step-size softmax policy gradient (SPG) (Agarwal et al., 2021; Mei et al., 2020) can only achieve sublinear convergence rates even with Nesterov acceleration (Chen et al., 2023). Hence, by changing the mirror map (from Euclidean to log-sum-exp) while using the same policy parameterization, **SPMA** can result in an exponential improvement over **SPG**.

Contribution 2: In order to handle MDPs with large state-action spaces, we use function approximation (e.g. linear models or neural networks) to parameterize the policies resulting in the class of log-linear or energy-based policies (Haarnoja et al., 2017; Agarwal et al., 2021; Yuan et al., 2023) respectively. By interpreting the policy parameterization as a constraint on the corresponding logits, we use projected mirror ascent to extend **SPMA** to the FA setting and design Algorithm 1. Unlike that for **NPG**, generalizing **SPMA** does not require compatible function approximation, and thus results in a more practical algorithm. Unlike **MDPO** (Tomar et al., 2020) which results in non-convex surrogates for linear FA, **SPMA** requires solving convex softmax classification problems in each iteration.

Contribution 3: In Section 4.2, we state the conditions under which Algorithm 1 converges to the neighbourhood of the optimal value function, and characterize the resulting linear convergence rate. Hence, for log-linear policies, Algorithm 1 matches the theoretical convergence of **NPG** with compatible function approximation (Agarwal et al., 2021; Alfano and Rebeschini, 2022; Yuan et al., 2023). Our theoretical results are better than those in Vaswani et al. (2021) and Schulman (2015) which prove sublinear convergence to a stationary point for idealized variants of **SPMA** and **TRPO** respectively. In contrast to Kuba et al. (2022) which prove that the idealized variants of **PPO** and **TRPO** converge to the optimal policy asymptotically, we characterize the non-asymptotic convergence rate for Algorithm 1.

Contribution 4: We empirically evaluate **SPMA** across simple MDPs with tabular and linear parameterization, Atari games with a discrete action space and a neural

policy parameterization with CNNs, and continuous control MuJoCo tasks with a continuous action space and a neural policy parameterization with MLPs. We demonstrate that **SPMA** has consistently good performance – on Atari games **SPMA** achieves better results than both **TRPO** and **PPO** while matching or outperforming **MDPO**, whereas on MuJoCo tasks, **SPMA** outperforms **PPO** and achieves similar or better results than **MDPO**.

2 PROBLEM FORMULATION

We consider an infinite-horizon discounted Markov decision process (MDP) (Puterman, 2014) defined by $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, r, \rho, \gamma \rangle$, where \mathcal{S} and \mathcal{A} represent the states and actions, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \rightarrow \Delta_{\mathcal{S}}$ is the transition probability function, $r : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ is the reward function, $\rho \in \Delta_{\mathcal{S}}$ is the initial state distribution, and $\gamma \in [0, 1)$ represents the discount factor. In this paper, we exclusively consider the setting where the number of states and actions is finite, but potentially large.

Given $s \in \mathcal{S}$, the policy π induces a probability distribution $\pi(\cdot|s)$ over the actions. The action-value function $Q^{\pi} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ induced by π is defined as $Q^{\pi}(s, a) := \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s, a_0 = a]$ where $s_t \sim p(\cdot|s_{t-1}, a_{t-1})$, and $a_t \sim \pi(\cdot|s_t)$ for $t \geq 1$. The value function corresponding to Q^{π} starting from state s is defined as $V^{\pi}(s) := \mathbb{E}_{a \sim \pi(\cdot|s)}[Q^{\pi}(s, a)]$ with $J(\pi) := V^{\pi}(\rho) := \mathbb{E}_{s \sim \rho}[V^{\pi}(s)]$ representing the expected discounted cumulative reward. Furthermore, the advantage function $A^{\pi} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is defined as $A^{\pi}(s, a) := Q^{\pi}(s, a) - V^{\pi}(s)$. The policy also induces a discounted state-occupancy measure $d^{\pi}(s) := (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t \Pr^{\pi}[s_t = s | s_0 \sim \rho]$ over the states. The objective is to find an optimal policy π^* that maximizes the expected reward $J(\pi)$, i.e. $\pi^* = \arg \max_{\pi} J(\pi)$. As a special case, in the bandit setting, $|\mathcal{S}| = 1$, $|\mathcal{A}| = K$, $\gamma = 0$, and $J(\pi) = \langle \pi, r \rangle$, with K representing the number of arms.

3 SOFTMAX POLICY MIRROR ASCENT: TABULAR PARAMETRIZATION

Softmax policy mirror ascent (referred to as **SPMA**) represents the policy using the softmax function $h : \mathbb{R}^A \rightarrow \Delta_A$ i.e. $\pi(\cdot|s) = h(z(s, \cdot))$ s.t. for all $(s, a) \in \mathcal{S} \times \mathcal{A}$, $\pi(a|s) = \frac{\exp(z(s, a))}{\sum_{a'} \exp(z(s, a'))}$, where the logits z are SA -dimensional vectors and Δ_A is the A -dimensional simplex. We first focus on the *tabular parameterization* where the number of parameters scales with the number of states and actions, and aim to learn the logits corresponding to an optimal policy. With some abuse of notation, we use $J(z)$ to refer to $J(\pi)$ where $\pi(\cdot|s) = h(z(s, \cdot))$ and state the objective as: $\max_{z \in \mathbb{R}^{SA}} J(z)$.

As the name suggests, **SPMA** uses mirror ascent (MA)

to maximize $J(z)$. For a differentiable, strictly convex mirror map Φ , MA (Beck and Teboulle, 2003; Bubeck et al., 2015) is an iterative algorithm whose update at iteration $t \in [T]$ can be stated in two equivalent ways:

$$\begin{aligned} \nabla\Phi(z_{t+1}) &= \nabla\Phi(z_t) + \eta \nabla_z J(z_t) \\ z_{t+1} &= \arg \max_{z \in \mathbb{R}^{SA}} \left[\langle z - z_t, \nabla_z J(z_t) \rangle - \frac{1}{\eta} D_\Phi(z, z_t) \right] \end{aligned} \quad (1)$$

where z_t is the logit at iteration t , η is the step-size and $D_\Phi(z, z') := \Phi(z) - \Phi(z') - \langle \nabla\Phi(z'), z - z' \rangle$ is the Bregman divergence between logits z and z' induced by the mirror map Φ . Hence, the MA update at iteration t can be interpreted as moving in the gradient direction $\nabla_z J(z_t)$ while staying “close” to the logit z_t , where the proximity between logits is measured according to the Bregman divergence and weighted by $\frac{1}{\eta}$.

3.1 Bandit Setting

It is instructive to first instantiate the SPMA update for the bandit setting where $J(\pi) = \langle \pi, r \rangle$. In this setting, $\nabla_z J(z) \in \mathbb{R}^A$ s.t. $[\nabla_z J(z)](a) = \pi(a) [r(a) - \langle \pi, r \rangle]$. Following Vaswani et al. (2021), we use the log-sum-exp mirror map i.e. $\phi(z) = \ln(\sum_a \exp(z(a)))$. Since $[\nabla\phi(z)](a) = \frac{\exp(z(a))}{\sum_{a'} \exp(z(a'))} = [h(z)](a) = \pi(a)$, the logit and probability spaces are dual to each other, and the $\nabla\phi$ map can be used to move between these spaces. Given this, the SPMA update can be written as:

$$\begin{aligned} \pi_{t+1}(a) &= \pi_t(a) (1 + \eta [r(a) - \langle \pi, r \rangle]) \\ &= \pi_t(a) [1 + \eta \sum_{a' \neq a} \pi_t(a') \Delta(a, a')], \end{aligned} \quad (2)$$

where $\Delta(a, a') := r(a) - r(a')$ represents the reward gap between arms a and a' . We first ensure that π_{t+1} is a valid probability distribution. Since $r(a) \in [0, 1]$ for all a , $\eta \leq 1$ is sufficient to guarantee that $\pi_{t+1}(a)$ is non-negative for every a . Moreover, $\sum_a \pi_{t+1}(a) = \sum_a \pi_t(a) + \eta \sum_a \pi_t(a) [r(a) - \langle \pi, r \rangle] = \sum_a \pi_t(a) = 1$. Hence, for $\eta \leq 1$, Eq. (2) results in a valid update to the policy. The above update is related to the PROD algorithm (Cesa-Bianchi et al., 2007) used for the experts problem in the online learning literature. In contrast to SPMA which is derived from mirror ascent, PROD is derived using a linearization of the Hedge (Freund and Schapire, 1997) algorithm and requires explicit normalization to obtain probabilities.

3.2 MDP Setting

In order to extend SPMA to the MDP setting, we use a (state-wise) weighted log-sum-exp mirror map, i.e. for a logit $z \in \mathbb{R}^{SA}$, we define $\Phi(z) := \sum_s w(s) \phi(z(s, \cdot)) = \sum_s w(s) \ln(\sum_a \exp(z(s, a)))$ where $w(s)$ are the per-state weights. Following the proof of Vaswani et al. (2024, Lemma 11), the resulting Bregman divergence is given as: $D_\Phi(z, z') = \sum_s w(s) \text{KL}(\pi'(\cdot|s) \parallel \pi(\cdot|s))$ where π and π' are the policies corresponding to logits z

and z' . At iteration t of SPMA, we choose $w(s) = d^{\pi_t}(s)$ and use the policy gradient theorem (Sutton et al., 1999) to calculate $[\nabla J(z_t)](s, a) = d^{\pi_t}(s) \pi_t(a|s) A^{\pi_t}(s, a)$. The resulting SPMA update is given as:

$$\begin{aligned} z_{t+1} &= \arg \max_{z \in \mathbb{R}^{SA}} \sum_s d^{\pi_t}(s) \left[\langle \pi_t(\cdot|s) A^{\pi_t}(s, \cdot), z(s, \cdot) \rangle \right. \\ &\quad \left. - \frac{1}{\eta} \text{KL}(\pi_t(\cdot|s) \parallel h(z(s, \cdot))) \right]. \end{aligned}$$

Since the above maximization decomposes over the states, we can write the per-state update for each $s \in \mathcal{S}$ in terms of $\pi_{t+1}(\cdot|s) = h(z_{t+1}(s, \cdot))$ as follows:

$$\pi_{t+1}(a|s) = \pi_t(a|s) (1 + \eta A^{\pi_t}(s, a)). \quad (3)$$

Similar to the bandit case, since $r(s, a) \in [0, 1]$, $\pi_{t+1}(a|s)$ is non-negative for $\eta \leq 1 - \gamma$. Since $\sum_a \pi_t(a|s) A^{\pi_t}(s, a) = 0$, $\sum \pi_{t+1}(a|s) = 1$, and hence Eq. (3) results in a valid policy update.

In order to compare the SPMA update to existing methods, note that for the tabular parameterization, natural policy gradient (NPG) update (Kakade, 2001) is the same as policy mirror ascent (Lan, 2023; Johnson et al., 2023; Xiao, 2022) and is given by: $\pi_{t+1}(a|s) \propto \pi_t(a|s) \exp(\eta A^{\pi_t}(s, a))$. In contrast to NPG, the SPMA update in Eq. (3) is linear in both η and $A^{\pi_t}(s, a)$ and does not require an explicit normalization across actions to ensure valid probability distributions. On the other hand, softmax policy gradient (SPG) (Agarwal et al., 2021; Mei et al., 2020) corresponds to choosing the mirror map ϕ in Eq. (1) to be the Euclidean norm and has the following update: $z_{t+1}(s, a) = z_t(s, a) + \eta \pi_t(a|s) A^{\pi_t}(s, a)$. Compared to SPG that uses the softmax policy gradient to update the logits, SPMA uses the softmax policy gradient to directly update the probabilities. As we demonstrate in the next section, this desirable property enables SPMA to achieve faster rates than SPG.

3.3 Theoretical Results

In this section, we prove convergence guarantees for SPMA in the multi-armed bandit and tabular MDP settings. We first establish linear convergence for SPMA for multi-armed bandits for any constant $\eta \leq 1$.

Theorem 1. *The SPMA update in Eq. (2) with (i) a constant step-size $\eta \leq 1$, and (ii) uniform initialization i.e. $\pi_0(a) = \frac{1}{K}$ for all a converges as:*

$$r(a^*) - \langle \pi_T, r \rangle \leq \left(1 - \frac{1}{K}\right) \exp\left(\frac{-\eta \Delta_{\min} T}{K}\right),$$

where T is the number of iterations, a^* is the optimal arm i.e. $a^* = \arg \max_a r(a)$ and $\Delta_{\min} := \min_{a \neq a^*} \Delta(a^*, a) = r(a^*) - r(a)$ is the gap.

The above theorem (proved in Appendix A) shows that for multi-armed bandit problems, SPMA can achieve linear convergence to the optimal arm, and the resulting rate depends on both the gap and the number of arms. In Appendix A.1, we prove that SPMA with specific gap-dependent step-sizes can achieve a global super-linear convergence rate for multi-armed bandits. To the best of our knowledge, these are the first global super-linear rates for PG methods on multi-armed bandit problems.

In the next theorem, we extend the linear convergence result to tabular MDPs and prove that when given access to the exact policy gradients, SPMA results in linear convergence to the optimal value function for any sufficiently small constant step-size.

Theorem 2. *Using the SPMA update in Eq. (3) with a step-size $\eta < \min \left\{ 1 - \gamma, \frac{1}{C_t(1-\gamma)} \right\}$ converges as:*

$$\left\| V^{\pi^*} - V^{\pi_T} \right\|_{\infty} \leq \left(\prod_{t=0}^{T-1} \alpha_t \right) \left\| V^{\pi^*} - V^{\pi_0} \right\|_{\infty},$$

where $\alpha_t := (1 - \eta C_t(1 - \gamma))$, $C_t := \min_s \{ \pi_t(\tilde{a}_t(s)|s) \Delta^t(s) \}$, $\tilde{a}_t(s) := \arg \max_a Q^{\pi_t}(s, a)$ and $\Delta^t(s) := \max_a Q^{\pi_t}(s, a) - \max_{a \neq \tilde{a}_t} Q^{\pi_t}(s, a)$.

For ease of exposition, the above theorem considers $\tilde{a}_t(s)$ to be the unique action maximizing $Q^{\pi_t}(s, \cdot)$ for every state s and policy π_t . In Appendix B, we extend this to include multiple optimal actions with a minor change in the definition of the gap. For rewards in $(0, 1)$, $C_t(1 - \gamma)$ is in $(0, 1)$ and depends on the initialization. If $C := \min_{t \in [T]} C_t$, then the above implies that when $T \in O\left(\frac{1}{\eta C(1-\gamma)} \ln(1/\epsilon)\right)$, SPMA guarantees that $V^{\pi_T}(s) \geq V^*(s) - \epsilon$ for all $s \in \mathcal{S}$. Note that in order to establish linear convergence, it is crucial to ensure that C is nonzero. Proving this property theoretically is challenging, as reflected in related work. In particular, the best-known linear convergence rates for NPG with a constant step size also depend on this same constant (see, for example, Theorem 5.4 in (Liu et al., 2024) and Lemma 10 in (Mei et al., 2021a)). Consequently, in Appendix C.3, we empirically verify that C is lower-bounded by a positive constant.

In order to put the above convergence result in context, note that SPG with a constant step-size results in a $\Theta(1/\epsilon)$ convergence (Mei et al., 2020). Recently, Chen et al. (2023) proved that constant step-size SPG with Nesterov acceleration can obtain an $O(1/\sqrt{\epsilon})$ convergence. In contrast, the above theorem demonstrates that by choosing the appropriate mirror map, constant step-size SPMA can achieve a faster $O(\log(1/\epsilon))$ rate of convergence. On the other hand, Liu et al. (2024); Lu et al. (2024) prove that SPG with adaptive step-sizes can also result in linear convergence. However, the

resulting rate depends on the distribution mismatch ratio $\left\| \frac{d^{\pi^*}}{\rho} \right\|_{\infty}$ that can be exponentially large in the size of the state space (Li et al., 2021). In contrast, the convergence result in Theorem 2 has no such dependence. The linear convergence rate in Theorem 2 matches that of NPG with a constant step-size (Liu et al., 2024) and compared to Liu et al. (2024, Theorem 5.4), it results in a better dependence (exponential vs polynomial) on the gap $\Delta^t(s)$. Finally, we note that for the tabular parameterization, a variant of TRPO has been shown to achieve $O(1/\epsilon^2)$ convergence to the optimal policy (Shani et al., 2020).

In the next section, we extend SPMA to exploit function approximation to handle large state-action spaces.

4 HANDLING FUNCTION APPROXIMATION

Handling large MDPs requires function approximation (FA) techniques to share information between states and actions. For example, given a set of state-action features $\mathbf{X} \in \mathbb{R}^{SA \times d}$ where $d \ll SA$, the *log-linear policy parameterization* (Agarwal et al., 2021; Alfano and Rebeschini, 2022; Yuan et al., 2023) considers policies of the form: $\pi(a|s) = \frac{\exp(\langle \mathbf{X}(s, a), \theta \rangle)}{\sum_{a'} \exp(\langle \mathbf{X}(s, a'), \theta \rangle)}$ where $\theta \in \mathbb{R}^d$ is the parameter to be learned. Hence, the log-linear policy parameterization can handle large state-action spaces while learning a compressed d -dimensional representation. We interpret the log-linear

Algorithm 1: SPMA with function approximation

Input: θ_0 (parameters for the initial policy π_0), f_{θ} (function approximation), T (number of outer-loop), m (number of inner-loops), η (outer-loop step-size), ζ (inner-loop step-size)

for $t \leftarrow 0$ **to** $T - 1$ **do**

1. Interact with the environment using π_t and form the surrogate function $\ell_t(\theta)$ in Eq. (5)

2. Initialize inner-loop: $\omega_0 = \theta_t$

for $k \leftarrow 0$ **to** $m - 1$ **do**

$\omega_{k+1} = \omega_k - \zeta \nabla_{\omega} \ell_t(\omega_k)$

end

3. $\theta_{t+1} = \omega_m$

4. Update $\pi_{t+1}(\cdot|s) = h(f_{\theta_{t+1}}(s, \cdot))$

end

Return θ_T

policy parameterization as a constraint in the space of logits. Specifically, the logits z are constrained to lie in the set $\mathcal{Z} = \{z \in \mathbb{R}^{SA} | \exists \theta \text{ s.t. } z = \mathbf{X}\theta\}$, meaning that the logits are required to be realizable by the linear model with features \mathbf{X} . We define Π as the corresponding set of feasible policies, i.e. $\Pi = \{\pi | \forall s \in \mathcal{S}, \pi(\cdot|s) = h(z(s, \cdot)) \text{ s.t. } z \in \mathcal{Z}\}$. Hence, the policies in Π are constrained to be log-linear. Note that, as in the case of log-linear policies, Π can be a

non-convex set, even when \mathcal{Z} is convex. For general *energy-based or neural policies* (Haarnoja et al., 2017; Agarwal et al., 2021), $\pi(a|s) \propto \exp(f_\theta(s, a))$ where $f_\theta : \mathbb{R}^{SA} \rightarrow \mathbb{R}$ is a complex, non-linear model. In this case, the logits are constrained to lie in the set: $\mathcal{Z} = \{z \in \mathbb{R}^{SA} | \exists \theta \text{ s.t. } z(s, a) = f_\theta(s, a)\}$.

The above interpretation allows us to extend SPMA to the FA setting. Specifically, we use the same mirror ascent update as in Eq. (1) with an additional projection step onto the feasible set \mathcal{Z} . Specifically, we define $z_{t+1/2}$ s.t. $\nabla \Phi(z_{t+1/2}) = \nabla \Phi(z_t) + \eta \nabla_z J(z_t)$ and compute $z_{t+1} = \arg \min_{z \in \mathcal{Z}} D_\Phi(z, z_{t+1/2})$. This step denotes the Bregman projection of $z_{t+1/2}$ onto \mathcal{Z} , i.e. we seek to find the closest (according to the Bregman divergence) realizable point (in \mathcal{Z}) to the “ideal” point $z_{t+1/2}$ which corresponds to using the tabular parameterization. Following Vaswani et al. (2021); Lavington et al. (2023), we convert the above projection problem into an unconstrained minimization problem where $\forall (s, a), z_{t+1}(s, a) = f_{\theta_{t+1}}(s, a), z_\theta(s, a) := f_\theta(s, a) \in \mathcal{Z}, \theta_{t+1} = \arg \min_\theta D_\Phi(z_\theta, z_{t+1/2})$ i.e. we aim to find the parameter θ that realizes the point $z_\theta \in \mathcal{Z}$ which is closest to $z_{t+1/2}$. Following Section 3.2, using the log-sum-exp mirror map weighted by $d^{\pi_t}(s)$ at iteration t results in the following optimization problem $\theta_{t+1} = \arg \min_\theta \tilde{\ell}_t(\theta)$ where,

$$\begin{aligned} \tilde{\ell}_t(\theta) &:= \sum_s d^{\pi_t}(s) \text{KL}(\pi_{t+1/2}(\cdot|s) || \pi_\theta(\cdot|s)) \\ &= \mathbb{E}_{s \sim d^{\pi_t}} \mathcal{H}(h(f_{\theta_t}(s, \cdot))(1 + \eta A^{\pi_t}(s, \cdot)), h(f_\theta(s, \cdot))) + C_t. \end{aligned} \quad (4)$$

Here, $\mathcal{H}(p, q) := -\mathbb{E}_p[\ln(q)] = -\sum_a p(a) \ln(q(a))$ is the cross-entropy between distributions p and q and C_t is a constant with respect to θ . We refer to $\tilde{\ell}_t(\theta)$ as the *ideal surrogate*. Minimizing this surrogate requires calculating the expectation over the states sampled according to π_t . In order to have a practical algorithm, we can run trajectories τ starting from the initial state distribution ρ , following the policy π_t and thus sampling from the d^{π_t} distribution (see Agarwal et al. (2021, Algorithm 3) for the detailed procedure). Given these samples, we form the surrogate $\ell_t(\theta)$ defined as:

$$\sum_{s \sim \tau} \text{KL}(h(f_{\theta_t}(s, \cdot))(1 + \eta A^{\pi_t}(s, \cdot)) || h(f_\theta(s, \cdot))) . \quad (5)$$

Note that $\mathbb{E}[\ell_t(\theta)] = \tilde{\ell}_t(\theta)$ where the expectation is w.r.t. to d^{π_t} . We use m steps of (stochastic) gradient descent to approximately minimize $\ell_t(\theta)$. Putting everything together, the algorithm incorporating general FA is presented in Algorithm 1.

Log-linear Policy Parameterization: For this special case, the problem in Eq. (4) is equivalent to a weighted (according to $d^{\pi_t}(s)$) multi-class classification for each state. The per-state problem corresponds

to a softmax classification into A classes using a linear model with features \mathbf{X} and soft labels equal to $\pi_{t+1/2}(\cdot|s)$. Computing θ_{t+1} thus involves minimizing a smooth, convex function.

In the next section, we compare Algorithm 1 to existing approaches that incorporate FA.

4.1 Comparison to Existing Approaches

Comparison to NPG: A principled extension of NPG to handle FA is via the compatible function approximation (Kakade, 2001; Agarwal et al., 2021). An example of such an algorithm, Q-NPG involves solving a quadratic surrogate at each iteration t : $\hat{\omega}_t = \min_\omega \sum_s d^{\pi_t}(s) \sum_a \pi_t(a|s) (f_\omega(s, a) - Q^{\pi_t}(s, a))^2$. The policy parameters are updated using $\hat{\omega}_t$ which corresponds to the natural gradient direction. While this approach results in theoretical guarantees (see Section 4.2 for details); for a general parameterization, the resulting algorithm involves changing the representation of the critic at every iteration. Consequently, solving the surrogate is expensive, limiting the practicality of the method.

Comparison to MDPO: A more practical extension of NPG is mirror descent policy optimization (Tomar et al., 2020) (MDPO). Similar to SPMA, MDPO can be interpreted as projected (onto the feasible set of policies) mirror ascent in the space of probabilities (Vaswani et al., 2021). The resulting surrogate (as a function of the policy parameters) is given by: $\sum_s d^{\pi_t}(s) \text{KL}(\pi_\theta(\cdot|s) || h(f_{\theta_t}(s, \cdot) \exp(\eta Q^{\pi_t}(s, \cdot))))$. Unlike the surrogate in Eq. (4), the MDPO surrogate is non-convex even when using a tabular softmax parameterization for the policy, and consequently does not have any theoretical guarantees. However, MDPO results in good empirical performance, and we compare to it in Section 5.

Comparison to TRPO: As explained in Vaswani et al. (2021), the surrogate in Eq. (4) is closely related to TRPO. In particular, the TRPO update consists of solving the following optimization problem: $\sum_s d^{\pi_t}(s) \sum_a \pi_t(a|s) A^{\pi_t}(s, a) \frac{\pi_\theta(a|s)}{\pi_{\theta_t}(a|s)}$, such that $\mathbb{E}_{s \sim d^{\pi_t}} [\text{KL}(\pi_t(\cdot|s) || \pi_\theta(\cdot|s))] \leq \delta$. SPMA (i) uses instead $\sum_s d^{\pi_t}(s) \sum_a \pi_t(a|s) A^{\pi_t}(s, a) \log \frac{\pi_\theta(a|s)}{\pi_{\theta_t}(a|s)}$, i.e. the logarithm of the importance sampling ratio, making the resulting update more stable (Vaswani et al., 2021) and (ii) enforces the proximity between policies via a regularization rather than a constraint. Enforcing the trust-region constraint in TRPO requires additional hyper-parameters, code-level optimizations and computation (Engstrom et al., 2019). In contrast, SPMA is more computationally efficient and simpler to implement in practice. In the next section, we study the theoretical properties of Algorithm 1.

4.2 Theoretical Guarantee

For rewards in $[0, 1]$ and for a general policy parameterization, Vaswani et al. (2021) prove that, for $\eta \leq 1 - \gamma$, Algorithm 1 results in monotonic improvement, i.e. $J(\pi_{t+1}) \geq J(\pi_t)$ and hence converges to a stationary point at an $O(1/\epsilon)$ rate. Since J is non-convex and can have multiple stationary points, the result in Vaswani et al. (2021) does not provide sufficient evidence for the good empirical performance of Algorithm 1. In this section, we prove that, under reasonable assumptions similar to existing works, Algorithm 1 can converge to the neighbourhood of the optimal value function at a linear rate. The size of the neighbourhood is determined by various practical factors such as sampling, inexact optimization, and bias due to the FA. In order to state our result, we first state and justify our assumptions.

Recall that in order to have a practical algorithm, we minimize $\ell_t(\theta)$ obtained by sampling from d^{π_t} .

Assumption 1. *Excess Risk:* For all iterations t of Algorithm 1, $|\tilde{\ell}_t(\theta_{t+1}) - \min \tilde{\ell}_t(\theta)| \leq \epsilon_{\text{stat}}$.

The above assumption quantifies the excess risk incurred by minimizing a finite sampled “dataset” of states as compared to minimizing over the population loss $\tilde{\ell}_t(\theta)$. This is a standard assumption in the literature analyzing the convergence of policy gradient methods with FA (Agarwal et al., 2021; Alfano and Rebeschini, 2022; Yuan et al., 2023). If n is the number of samples and the surrogate is minimized using (stochastic) gradient descent, using the standard generalization results (Lei and Ying, 2021; Nikolakakis et al., 2022), we expect $\epsilon_{\text{stat}} = O(1/n)$ for the log-linear parameterization. For example, using m iterations of SGD would result in $\epsilon_{\text{stat}} = O(1/n + 1/m)$ (Lei and Ying, 2021, Theorem 6). For a general parameterization, where the surrogate might be non-convex, the excess risk can be bounded up to the optimization error (Nicolakakis et al., 2022). Under the appropriate technical assumptions, $\ell_t(\theta)$ can be shown to satisfy the Polyak-Lojasiewicz condition (Liu et al., 2022) implying that the optimization error for (stochastic) gradient descent can be made arbitrarily small. The next assumption quantifies the bias incurred because of a policy parameterization with limited expressive power compared to using the tabular parameterization.

Assumption 2. *Bias:* For all iterations t of Algorithm 1, $\min_{\theta} \tilde{\ell}_t(\theta) \leq \epsilon_{\text{bias}}$.

The above assumption captures the flexibility of the model class being used in the policy parameterization. For a tabular parameterization where the number of parameters scales as SA , $\epsilon_{\text{bias}} = 0$, whereas for the log-linear parameterization, ϵ_{bias} depends on the expressivity of the features. The final assumption is

concerned with exploration and indicates that the initial state distribution has full support implying that the method does not require explicit exploration.

Assumption 3. *Exploration:* $\forall s \in \mathcal{S}, \rho(s) \geq \rho_{\min} > 0$.

The above assumption is standard in the literature (Agarwal et al., 2021; Xiao, 2022) and helps isolate and study the optimization properties of PG methods. We prove the following theorem in Appendix B.

Theorem 3. *Under assumption 1-3, Algorithm 1 with $\eta < \min \left\{ 1 - \gamma, \frac{1}{C_t(1-\gamma)} \right\}$ converges as,*

$$\begin{aligned} & J(\pi^*) - J(\pi_T) \\ & \leq \left(\prod_{t=0}^{T-1} \alpha_t \right) (J(\pi^*) - J(\pi_0)) + \beta \sum_{t=0}^{T-1} \prod_{i=t+1}^{T-1} \alpha_i, \end{aligned}$$

where $\beta = \frac{\sqrt{2}}{(1-\gamma)^2 \rho_{\min}} \sqrt{\epsilon_{\text{stat}} + \epsilon_{\text{bias}}}$ and α_t has the same definition as in Theorem 2.

The above theorem shows that Algorithm 1 converges linearly to the neighbourhood of the optimal value function. Furthermore, for the log-linear parameterization, the size of the neighbourhood can be bounded explicitly. For example, if the logits $z_{t+1/2}$ for every t lie in the span of the features, $\epsilon_{\text{bias}} = 0$ (this is similar to the linear Bellman completeness condition used in the analysis of value-based methods (Munos, 2005)) and $\epsilon_{\text{stat}} = O(1/n + 1/m)$. By using more samples and with more (S)GD iterations, the size of the neighbourhood can be made arbitrarily small. Except for the neighbourhood term, the above convergence result is similar to that for the tabular setting in Theorem 2. The other difference is that the result in the tabular setting holds in the ℓ_{∞} norm and thus holds for all states, whereas the result in Theorem 3 only holds for a fixed starting state distribution ρ . In practice, A^{π_t} is typically estimated via a critic. To account for this, we generalize the proof of Theorem 3 in Appendix B, and prove that Algorithm 1 converges linearly to a neighbourhood that depends on an additional term proportional to the critic error.

We now compare to the existing theoretical results for PG methods with FA. For the log-linear policy parameterization, Q-NPG and its variants have been shown to achieve linear convergence to the neighbourhood of the optimal value function (Agarwal et al., 2021; Alfano and Rebeschini, 2022; Yuan et al., 2023). The size of the neighbourhood depends on similar quantities as Theorem 3. Finally, we note that while an ideal, impractical variant of TRPO has a monotonic improvement guarantee similar to Algorithm 1 (Schulman, 2015), it does not have convergence guarantees comparable to Theorem 3.

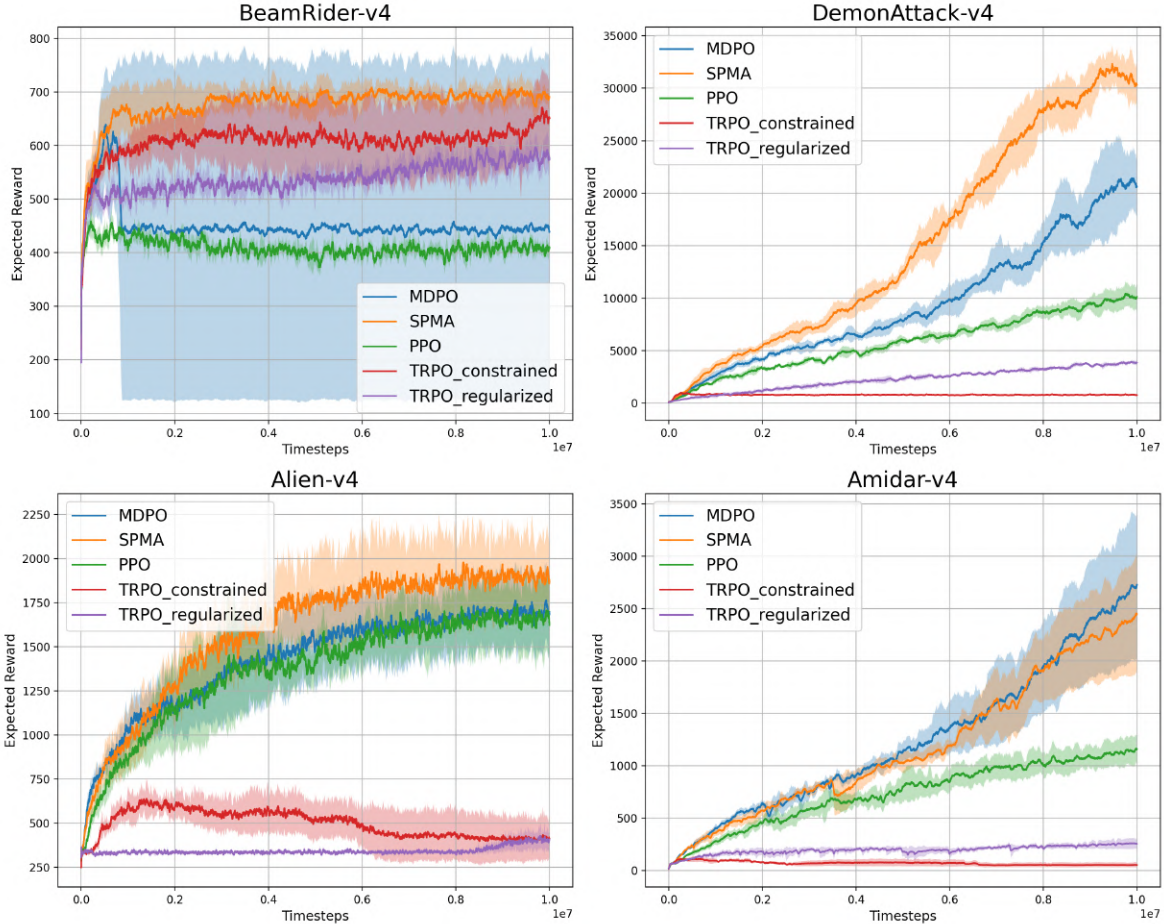


Figure 1: On Atari games, where a CNN-based actor network is employed, SPMA matches or surpasses MDPO and outperforms PPO, as well as both the constrained and regularized versions of TRPO.

5 EMPIRICAL EVALUATION

We evaluate SPMA¹ on three types of problems: (i) tabular MDPs with access to exact policy gradients, (ii) MDPs with continuous states but discrete actions, using inexact policy gradients, and (iii) MDPs with continuous state-actions spaces and inexact gradients. For tabular MDPs, we use the tabular parameterization and compare SPMA against NPG and constant step-size SPG (Mei et al., 2020). For these environments, we also consider log-linear policies and compare SPMA to MDPO and SPG. For non-tabular environments, we consider PPO, TRPO and MDPO as baselines. We consider two variants of TRPO – TRPO-constrained, the standard optimized variant in Raffin et al. (2021) and TRPO-regularized, the original regularized variant. TRPO-constrained is able to effectively enforce the trust region constraint using conjugate gradient, but introduces additional hyper-parameters, requires code-level optimization techniques and is computationally

expensive. On the other hand, TRPO-regularized is significantly more efficient and theoretically principled (Lazić et al., 2021), and is similar to SPMA’s objective (see Section 4.1). For details regarding the hyper-parameters of all methods for each environment, refer to Appendices C and D.

Tabular MDP Results: We present the results in Appendix C, and summarize the key findings here. We observe that SPMA and NPG achieve comparable performance, both consistently outperforming SPG (Figure 3). However, in the linear FA setting, SPMA demonstrates superior performance compared to MDPO in the CliffWorld environment (Sutton, 2018) (Figure 5), while performing similarly in the Frozen Lake environment (Brockman, 2016) (Figure 6). In both environments, SPMA and MDPO consistently outperform SPG.

In the remainder of this section, we focus on the non-tabular settings with inexact policy gradients. For these experiments, we follow the protocol of Tomar

¹The code is available at <http://github.com/reza-asad/SPMA>.

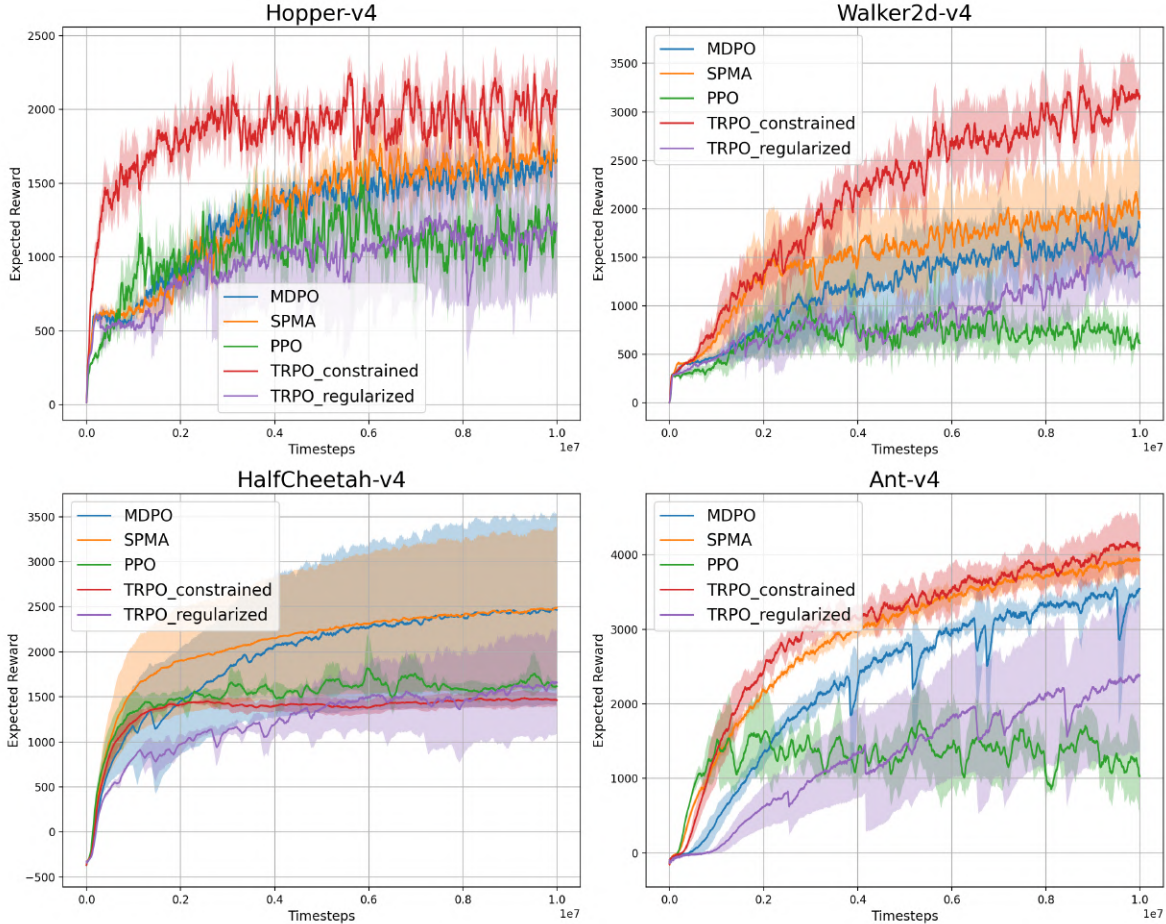


Figure 2: On MuJoCo control tasks, where a two-layer MLP actor network is used, **SPMA** matches or outperforms MDPO while consistently outperforming PPO and regularized TRPO. In contrast to the results on Atari games, with a shallow MLP, **TRPO-constrained** outperforms all methods.

et al. (2020), using 5 seeds and reporting the average results along with their 95% confidence intervals. Additionally, we employ the actor-critic architecture, policy parameterization, and GAE (Schulman et al., 2015) (to estimate the advantage function) from stable baselines (Raffin et al., 2021). We emphasize that, in contrast to prior work, we do not make ad-hoc adjustments to **SPMA** (i.e., the actor). To set the step-size η in Algorithm 1, we perform a grid search over five values (fixed across all experiments) and set the inner loop step-size ζ using Armijo line search (Armijo, 1966).

Atari and Mujoco Results: We evaluate the performance of **SPMA** compared to the baselines across various Atari 2600 games (Bellemare et al., 2013) and MuJoCo (Todorov et al., 2012) control tasks from OpenAI Gym (Brockman, 2016). The observation space for Atari games consists of a $210 \times 160 \times 3$ image, representing the current state of the game. The action space in these environments is discrete, whereas in MuJoCo, it is continuous and by default represented by a

diagonal Gaussian distribution in Raffin et al. (2021). Additionally, the actor-critic network for Atari uses a CNN as a feature extractor, while MuJoCo employs an MLP.

Comparing the results in Fig. 1 and 2, our key observations are as follows: i) **SPMA** consistently outperforms or matches MDPO and PPO across all environments; ii) although **TRPO-constrained** achieves superior performance on MuJoCo, its performance degrades considerably on Atari games. We conjecture that the conjugate gradient algorithm in **TRPO-constrained** performs poorly when the actor network is a CNN rather than a two-layer MLP; iii) **TRPO-regularized**, which has a similar objective as **SPMA** (see Section 4.1 for a comparison) does not perform as well on MuJoCo and has considerably worse performance on Atari. Hence, we observe that replacing the sampling ratio by its log can result in substantial empirical gains. This behaviour has also been observed for PPO Vaswani et al. (2021). Overall, our experiments demonstrate that,

despite being theoretically grounded, SPMA exhibits strong empirical performance across various environments without relying on ad-hoc adjustments.

6 DISCUSSION

We developed SPMA, a PG method that corresponds to mirror ascent in the dual space of logits. We believe that our paper bridges the gap between theoretical PG methods and practical objectives by presenting a method that offers strong theoretical convergence guarantees while delivering competitive practical performance (compared to PPO, TRPO, MDP0), without relying on additional heuristics or algorithmic modifications. In the future, we aim to develop techniques for adaptively tuning the step-size and avoiding expensive grid-searches. We also plan to develop and analyze an off-policy variant of SPMA.

References

- Agarwal, A., Kakade, S. M., Lee, J. D., and Mahajan, G. (2021). On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *J. Mach. Learn. Res.*, 22(98):1–76.
- Alfano, C. and Rebeschini, P. (2022). Linear convergence for natural policy gradient with log-linear policy parametrization. *arXiv preprint arXiv:2209.15382*.
- Armijo, L. (1966). Minimization of functions having lipschitz continuous first partial derivatives. *Pacific Journal of mathematics*, 16(1):1–3.
- Beck, A. and Teboulle, M. (2003). Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175.
- Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. (2013). The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279.
- Bhandari, J. and Russo, D. (2021). On the linear convergence of policy gradient methods for finite mdps. In *International Conference on Artificial Intelligence and Statistics*, pages 2386–2394. PMLR.
- Brockman, G. (2016). Openai gym. *arXiv preprint arXiv:1606.01540*.
- Bubeck, S. et al. (2015). Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357.
- Cesa-Bianchi, N., Mansour, Y., and Stoltz, G. (2007). Improved second-order bounds for prediction with expert advice. *Machine Learning*, 66:321–352.
- Chen, Y.-J., Huang, N.-C., Lee, C.-p., and Hsieh, P.-C. (2023). Accelerated policy gradient: On the convergence rates of the nesterov momentum for reinforcement learning. In *Forty-first International Conference on Machine Learning*.
- Engstrom, L., Ilyas, A., Santurkar, S., Tsipras, D., Janoos, F., Rudolph, L., and Madry, A. (2019). Implementation matters in deep rl: A case study on ppo and trpo. In *International conference on learning representations*.
- Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139.
- Haarnoja, T., Tang, H., Abbeel, P., and Levine, S. (2017). Reinforcement learning with deep energy-based policies. In *International conference on machine learning*, pages 1352–1361. PMLR.
- Johnson, E., Pike-Burke, C., and Rebeschini, P. (2023). Optimal convergence rate for exact policy mirror descent in discounted markov decision processes. *arXiv preprint arXiv:2302.11381*.
- Kakade, S. M. (2001). A natural policy gradient. *Advances in neural information processing systems*, 14.
- Khodadadian, S., Jhunjhunwala, P. R., Varma, S. M., and Maguluri, S. T. (2021). On the linear convergence of natural policy gradient algorithm. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 3794–3799. IEEE.
- Konda, V. R. and Tsitsiklis, J. N. (2000). Actor-critic algorithms. In *Advances in neural information processing systems*, pages 1008–1014.
- Kuba, J. G., de Witt, C. S., and Foerster, J. (2022). Mirror learning: A unifying framework of policy optimisation. *arXiv preprint arXiv:2201.02373*.
- Lan, G. (2023). Policy mirror descent for reinforcement learning: Linear convergence, new sampling complexity, and generalized problem classes. *Mathematical programming*, 198(1):1059–1106.
- Lavington, J. W., Vaswani, S., Babanezhad, R., Schmidt, M., and Roux, N. L. (2023). Target-based surrogates for stochastic optimization. *arXiv preprint arXiv:2302.02607*.
- Lazić, N., Hao, B., Abbasi-Yadkori, Y., Schuurmans, D., and Szepesvári, C. (2021). Optimization issues in kl-constrained approximate policy iteration. *arXiv preprint arXiv:2102.06234*.
- Lei, Y. and Ying, Y. (2021). Sharper generalization bounds for learning with gradient-dominated objective functions. In *International Conference on Learning Representations*.
- Li, G., Wei, Y., Chi, Y., Gu, Y., and Chen, Y. (2021). Softmax policy gradient methods can take exponen-

- tial time to converge. In *Conference on Learning Theory*, pages 3107–3110. PMLR.
- Liu, B., Cai, Q., Yang, Z., and Wang, Z. (1906). Neural proximal/trust region policy optimization attains globally optimal policy (2019). *arXiv preprint arXiv:1906.10306*.
- Liu, C., Zhu, L., and Belkin, M. (2022). Loss landscapes and optimization in over-parameterized non-linear systems and neural networks. *Applied and Computational Harmonic Analysis*, 59:85–116.
- Liu, J., Li, W., and Wei, K. (2024). Elementary analysis of policy gradient methods. *arXiv preprint arXiv:2404.03372*.
- Lu, M., Aghaei, M., Raj, A., and Vaswani, S. (2024). Towards principled, practical policy gradient for bandits and tabular mdps. *arXiv preprint arXiv:2405.13136*.
- Mei, J., Dai, B., Xiao, C., Szepesvari, C., and Schuurmans, D. (2021a). Understanding the effect of stochasticity in policy optimization. *Advances in Neural Information Processing Systems*, 34:19339–19351.
- Mei, J., Gao, Y., Dai, B., Szepesvari, C., and Schuurmans, D. (2021b). Leveraging non-uniformity in first-order non-convex optimization. In *International Conference on Machine Learning*, pages 7555–7564. PMLR.
- Mei, J., Xiao, C., Szepesvari, C., and Schuurmans, D. (2020). On the global convergence rates of softmax policy gradient methods. In *International Conference on Machine Learning*, pages 6820–6829. PMLR.
- Munos, R. (2005). Error bounds for approximate value iteration. In *Proceedings of the National Conference on Artificial Intelligence*, volume 20, page 1006. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.
- Nikolakakis, K. E., Haddadpour, F., Karbasi, A., and Kalogerias, D. S. (2022). Beyond lipschitz: Sharp generalization and excess risk bounds for full-batch gd. *arXiv preprint arXiv:2204.12446*.
- Puterman, M. L. (2014). *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.
- Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., and Dormann, N. (2021). Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8.
- Schulman, J. (2015). Trust region policy optimization. *arXiv preprint arXiv:1502.05477*.
- Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P. (2015). High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*.
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- Shani, L., Efroni, Y., and Mannor, S. (2020). Adaptive trust region policy optimization: Global convergence and faster rates for regularized mdps. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5668–5675.
- Sutton, R. S. (2018). Reinforcement learning: An introduction. *A Bradford Book*.
- Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. (1999). Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12.
- Todorov, E., Erez, T., and Tassa, Y. (2012). Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE.
- Tomar, M., Shani, L., Efroni, Y., and Ghavamzadeh, M. (2020). Mirror descent policy optimization. *arXiv preprint arXiv:2005.09814*.
- Vaswani, S., Bachem, O., Totaro, S., Müller, R., Garg, S., Geist, M., Machado, M. C., Castro, P. S., and Roux, N. L. (2021). A general class of surrogate functions for stable and efficient reinforcement learning. *arXiv preprint arXiv:2108.05828*.
- Vaswani, S., Kazemi, A., Babanezhad Harikandeh, R., and Le Roux, N. (2024). Decision-aware actor-critic with function approximation and theoretical guarantees. *Advances in Neural Information Processing Systems*, 36.
- Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.
- Xiao, L. (2022). On the convergence rates of policy gradient methods. *Journal of Machine Learning Research*, 23(282):1–36.
- Yuan, R., Du, S. S., Gower, R. M., Lazaric, A., and Xiao, L. (2023). Linear convergence of natural policy gradient methods with log-linear policies. In *International Conference on Learning Representations*.
- Zhong, H. and Zhang, T. (2024). A theoretical analysis of optimistic proximal policy optimization in linear markov decision processes. *Advances in Neural Information Processing Systems*, 36.

Checklist

1. For all models and algorithms presented, check if you include:
 - (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
 - (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
 - (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [No]
2. For any theoretical claim, check if you include:
 - (a) Statements of the full set of assumptions of all theoretical results. [Yes]
 - (b) Complete proofs of all theoretical results. [Yes]
 - (c) Clear explanations of any assumptions. [Yes]
3. For all figures and tables that present empirical results, check if you include:
 - (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
 - (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
 - (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
 - (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Not Applicable]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
 - (a) Citations of the creator If your work uses existing assets. [Yes]
 - (b) The license information of the assets, if applicable. [Not Applicable]
 - (c) New assets either in the supplemental material or as a URL, if applicable. [No]
 - (d) Information about consent from data providers/curators. [Not Applicable]
 - (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
 - (a) The full text of instructions given to participants and screenshots. [Not Applicable]
 - (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
 - (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Supplementary Material

Organization of the Appendix

A Multi-armed Bandit Proofs

B MDP Proofs

C Tabular MDP Experiments

D Additional Details for Stable Baselines Experiments

A Multi-armed Bandit Proofs

Theorem 1. *The SPMA update in Eq. (2) with (i) a constant step-size $\eta \leq 1$, and (ii) uniform initialization i.e. $\pi_0(a) = \frac{1}{K}$ for all a converges as:*

$$r(a^*) - \langle \pi_T, r \rangle \leq \left(1 - \frac{1}{K}\right) \exp\left(\frac{-\eta \Delta_{\min} T}{K}\right),$$

where T is the number of iterations, a^* is the optimal arm i.e. $a^* = \arg \max_a r(a)$ and $\Delta_{\min} := \min_{a \neq a^*} \Delta(a^*, a) = r(a^*) - r(a)$ is the gap.

Proof. . As in equation (2), we can write the update for arm a as following where $\Delta(a, a') = r(a) - r(a')$,

$$\begin{aligned} \pi_{t+1}(a) &= \pi_t(a) \left[1 + \eta \sum_{a' \neq a} \pi_t(a') \Delta(a, a') \right] \\ 1 - \pi_{t+1}(a^*) &= 1 - \pi_t(a^*) - \eta \pi_t(a^*) \left[\sum_{a' \neq a^*} \pi_t(a') \Delta(a^*, a') \right] \end{aligned} \tag{6}$$

We first find a lower-bound for $\sum_{a' \neq a^*} \pi_t(a') \Delta(a^*, a')$:

$$\begin{aligned} \sum_{a' \neq a^*} \pi_t(a') \Delta(a^*, a') &\geq \Delta_{\min} \sum_{a' \neq a^*} \pi_t(a') \\ &= \Delta_{\min} (1 - \pi_t(a^*)) \end{aligned} \tag{7}$$

Next, we observe that $\sum_{a' \neq a^*} \pi_t(a') \Delta(a^*, a') \geq 0$. Using this information and starting with a uniform initialization for selecting an arm implies a monotonic improvement on the probability of selecting the optimal arm:

$$\pi_{t+1}(a^*) > \pi_t(a^*) > \dots > \pi_0(a^*) = \frac{1}{K} \tag{8}$$

Let $\epsilon_t = 1 - \pi_t(a^*)$.

$$\begin{aligned}
 \epsilon_{t+1} &= \epsilon_t - \eta \pi_t(a^*) \left[\sum_{a' \neq a^*} \pi_t(a') \Delta(a^*, a') \right] \\
 &\leq \epsilon_t - \frac{\eta}{K} \left[\sum_{a' \neq a^*} \pi_t(a') \Delta(a^*, a') \right] && \text{(using (8))} \\
 &\leq \epsilon_t - \frac{\eta \Delta_{\min}}{K} \epsilon_t && \text{(using (7))} \\
 &= \epsilon_t \left(1 - \frac{\eta \Delta_{\min}}{K} \right)
 \end{aligned}$$

Recurring from $t = 0$ to $t = T - 1$ we have:

$$\begin{aligned}
 \epsilon_T &\leq \epsilon_0 \left(1 - \frac{\eta \Delta_{\min}}{K} \right)^T \\
 &\leq \epsilon_0 \exp \left(\frac{-\eta \Delta_{\min} T}{K} \right) && \text{(using } 1 - x \leq \exp(-x) \text{)} \\
 &= \left(1 - \frac{1}{K} \right) \exp \left(\frac{-\eta \Delta_{\min} T}{K} \right)
 \end{aligned}$$

Finally, we define the sub-optimality gap, $\delta_T := r(a^*) - \langle \pi_T, r \rangle$:

$$\begin{aligned}
 \delta_T &= \sum_{a'} \pi_T(a') [r(a^*) - r(a')] \\
 &= \sum_{a' \neq a^*} \pi_T(a') \Delta(a^*, a') \\
 &\leq \max_{a'} \Delta(a^*, a') \sum_{a' \neq a^*} \pi_T(a') \\
 &= \max_{a'} \Delta(a^*, a') (1 - \pi_T(a^*)) \\
 &\leq 1 - \pi_T(a^*) && \text{(using the fact } 0 \leq r \leq 1 \text{)} \\
 &= \epsilon_T \\
 &\leq \left(1 - \frac{1}{K} \right) \exp \left(\frac{-\eta \Delta_{\min} T}{K} \right)
 \end{aligned}$$

□

A.1 Super-linear Rate for Bandits

In order to achieve the desired fast rate of convergence, we modify the update in Eq. (2) to use a set of $\binom{K}{2}$ constant gap-dependent step-sizes $\{\eta_{a,a'}\}_{a,a' \in [K]}$. The new update can be written as:

$$\pi_{t+1}(a) = \pi_t(a) \left[1 + \sum_{a' \neq a} \pi_t(a') \eta_{a,a'} \Delta(a, a') \right] \quad (9)$$

The following theorem shows that the above update results in super-linear convergence.

Theorem 4. *Using the SPMA update in Eq. (9) with (i) $\eta_{a,a'} = \frac{1}{|\Delta(a,a')|}$ and a (ii) uniform initialization similar to Theorem 1 results in valid probability distributions and converges as:*

$$r(a^*) - \langle \pi_T, r \rangle \leq \left[\left(1 - \frac{1}{K} \right) \right]^{2^T}$$

where T is the number of iterations, a^* is the optimal arm and $\Delta(a, a') := r(a) - r(a')$ represents the reward gap between arms a and a' .

Proof. We define $\Delta(a, a') := r(a) - r(a')$,

$$\begin{aligned} A^{\pi_t} &= r(a) - \langle \pi_t, r \rangle \\ &= \sum_{a'} \pi_t(a') [r(a) - r(a')] \\ &= \sum_{a'} \pi_t(a') \Delta(a, a') \end{aligned}$$

Choosing different step sizes for every pair of arms, depending on their corresponding gap, $\eta_{a,a'} = \frac{1}{|\Delta(a,a')|}$ we get the following update for $\pi_{t+1}(a)$:

$$\begin{aligned} \pi_{t+1}(a) &= \pi_t(a) \left[1 + \sum_{a' \neq a} \eta_{a,a'} \pi_t(a') \Delta(a, a') \right] \\ &= \pi_t(a) \left[1 + \sum_{a' \neq a} \pi_t(a') \text{sign}(\Delta(a, a')) \right] \end{aligned} \quad (\text{i})$$

Now we check if π_{t+1} is a probability distribution with this choice of η . Note that $\Delta(a, a') = -\Delta(a', a)$.

$$\begin{aligned} \sum_a \pi_{t+1}(a) &= \sum_a \pi_t(a) + \sum_a \pi_t(a) \sum_{a' \neq a} \pi_t(a') \text{sign}(\Delta(a, a')) \\ &= 1 + \sum_{(a,a'), a \neq a'} \pi_t(a) \pi_t(a') (\text{sign}(\Delta(a, a')) + \text{sign}(\Delta(a', a))) \\ &= 1 + \sum_{(a,a'), a \neq a'} \pi_t(a) \pi_t(a') (\text{sign}(\Delta(a, a')) - \text{sign}(\Delta(a, a'))) \quad (\text{ since } \Delta(a, a') = -\Delta(a', a)) \\ &= 1 \end{aligned}$$

Furthermore, it is clear that $\pi_t(a) \in [0, 1]$. Based on this we just need to show that the probability of the optimal arm a^* converges to 1.

Computing the probability of pulling the optimal arm using update (i):

$$\begin{aligned} \pi_{t+1}(a^*) &= \pi_t(a^*) \left[1 + \sum_{a' \neq a^*} \pi_t(a') \text{sign}(\Delta(a^*, a')) \right] \\ &= \pi_t(a^*) \left[1 + \sum_{a' \neq a^*} \pi_t(a') \right] \quad (\Delta(a^*, a') > 0 \quad \forall a') \\ &= \pi_t(a^*) [2 - \pi_t(a^*)] \end{aligned} \quad (\text{ii})$$

We use induction to show $\pi_t(a^*) = 1 - \left[\left(1 - \frac{1}{K} \right) \right]^{2^t}$ solves the recurrence relation (ii). We consider the uniform distribution over the arms at the initialization i.e. $\pi_0(a) = \frac{1}{K}$, $\forall a \in \mathcal{A}$. For the base case, we show the suggested solution satisfies recursion (ii):

$$\begin{aligned} \pi_1(a^*) &= \frac{1}{K} \left(2 - \frac{1}{K} \right) \quad (\text{using the recursion in (ii)}) \\ &= \left(1 - 1 + \frac{1}{K} \right) \left(1 + 1 - \frac{1}{K} \right) \\ &= 1 - \left[\left(1 - \frac{1}{K} \right) \right]^2 \end{aligned}$$

Assuming the suggested solution is true for t , we show it is also true for $t + 1$:

$$\begin{aligned}\pi_{t+1}(a^*) &= \left[1 - \left(1 - \frac{1}{K}\right)^{2^t}\right] \left[2 - 1 + \left(1 - \frac{1}{K}\right)^{2^t}\right] \\ &= 1 - \left[\left(1 - \frac{1}{K}\right)^{2^{t+1}}\right]\end{aligned}$$

Let $\delta_T := r(a^*) - \langle \pi_T, r \rangle$ represent the sub-optimality gap.

$$\begin{aligned}\delta_T &= \sum_{a'} \pi_T(a') [r(a^*) - r(a')] \\ &= \sum_{a' \neq a^*} \pi_T(a) \Delta(a^*, a) \\ &\leq \max_{a'} \Delta(a^*, a') \sum_{a' \neq a^*} \pi_T(a) \\ &\leq 1 - \pi_T(a^*) \quad \text{(using the fact } 0 \leq r \leq 1) \\ &= \left[\left(1 - \frac{1}{K}\right)\right]^{2^T} \quad \text{(using the formula for } \pi_T(a^*))\end{aligned}$$

□

B MDP Proofs

B.1 Tabular Setting

Lemma 1. *For any policy π_t we have*

$$\sum_a \pi_t(a|s) [A^{\pi_t}(s, a)]^2 \geq C_t \max_a A^{\pi_t}(s, a)$$

where $C_t := \min_s \{\pi_t(\tilde{\mathcal{A}}_t(s)|s) \Delta^t(s)\}$, $\tilde{\mathcal{A}}_t(s) := \arg \max_{a \in \mathcal{A}} Q^{\pi_t}(s, a)$, $\pi_t(\tilde{\mathcal{A}}_t(s)|s) = \sum_{a \in \tilde{\mathcal{A}}_t(s)} \pi_t(a_t(s)|s)$ and $\Delta^t(s) := \max_{a \in \mathcal{A}} Q^{\pi_t}(s, a) - \max_{a \notin \tilde{\mathcal{A}}_t(s)} Q^{\pi_t}(s, a)$.

Proof. Recall $\tilde{\mathcal{A}}_t(s) := \arg \max_{a \in \mathcal{A}} A^{\pi_t}(s, a)$ i.e. $\tilde{\mathcal{A}}_t(s)$ is a set containing actions with maximum advantage for state s . Let's define $\pi_t(\tilde{\mathcal{A}}_t(s)|s) = \sum_{a \in \tilde{\mathcal{A}}_t(s)} \pi_t(\tilde{a}_t(s)|s)$. We can split the sum on the LHS of the above over $\tilde{\mathcal{A}}_t(s)$:

$$\begin{aligned}\sum_a \pi_t(a|s) [A^{\pi_t}(s, a)]^2 &= \sum_{a \in \tilde{\mathcal{A}}_t(s)} \pi_t(\tilde{a}_t(s)|s) [\max_a A^{\pi_t}(s, a)] [\max_a A^{\pi_t}(s, a)] \\ &\quad + \sum_{a \notin \tilde{\mathcal{A}}_t(s)} \pi_t(a|s) [A^{\pi_t}(s, a)]^2 \\ &= \pi_t(\tilde{\mathcal{A}}_t(s)|s) [\max_a A^{\pi_t}(s, a)] [\max_a A^{\pi_t}(s, a)] \\ &\quad + \sum_{a \notin \tilde{\mathcal{A}}_t(s)} \pi_t(a|s) [A^{\pi_t}(s, a)]^2\end{aligned} \tag{10}$$

Let $\tilde{\pi}_t$ be the following distribution over the actions.

$$\tilde{\pi}_t(a|s) = \begin{cases} 0 & \text{if } a \in \tilde{\mathcal{A}}_t(s) \\ \frac{\pi_t(a|s)}{1 - \pi_t(\tilde{\mathcal{A}}_t(s)|s)} & \text{otherwise} \end{cases}$$

Re-writing $\sum_a \pi_t(a|s) A^{\pi_t}(s, a) = 0$ using the above distribution we obtain:

$$(1 - \pi_t(\tilde{\mathcal{A}}_t(s)|s)) \mathbb{E}_{a \sim \tilde{\pi}_t} [A^{\pi_t}(s, a)] + \pi_t(\tilde{\mathcal{A}}_t(s)|s) [\max_a A^{\pi_t}(s, a)] = 0$$

$$(1 - \pi_t(\tilde{\mathcal{A}}_t(s)|s)) \mathbb{E}_{a \sim \tilde{\pi}_t}[A^{\pi_t}(s, a)] = -\pi_t(\tilde{\mathcal{A}}_t(s)|s) [\max_a A^{\pi_t}(s, a)] \quad (11)$$

Expanding the second term in Eq. 10 using $\tilde{\pi}_t$ we obtain:

$$\begin{aligned} \sum_{a \notin \tilde{\mathcal{A}}_t(s)} \pi_t(a|s) [A^{\pi_t}(s, a)]^2 &= (1 - \pi_t(\tilde{\mathcal{A}}_t(s)|s)) \mathbb{E}_{a \sim \tilde{\pi}_t}[A^{\pi_t}(s, a)]^2 \\ &\geq (1 - \pi_t(\tilde{\mathcal{A}}_t(s)|s)) (\mathbb{E}_{a \sim \tilde{\pi}_t}[A^{\pi_t}(s, a)])^2 \quad (\text{using } \mathbb{E}[x^2] \geq (\mathbb{E}[x])^2) \\ &= (1 - \pi_t(\tilde{\mathcal{A}}_t(s)|s)) (\mathbb{E}_{a \sim \tilde{\pi}_t}[A^{\pi_t}(s, a)]) (\mathbb{E}_{a \sim \tilde{\pi}_t}[A^{\pi_t}(s, a)]) \\ &= -\pi_t(\tilde{\mathcal{A}}_t(s)|s) [\max_a A^{\pi_t}(s, a)] (\mathbb{E}_{a \sim \tilde{\pi}_t}[A^{\pi_t}(s, a)]) \quad (\text{using Eq. 11}) \end{aligned}$$

Plugging in the result above into Eq. 10 we obtain:

$$\begin{aligned} \sum_a \pi_t(a|s) [A^{\pi_t}(s, a)]^2 &\geq \pi_t(\tilde{\mathcal{A}}_t(s)|s) [\max_a A^{\pi_t}(s, a)] [\max_a A^{\pi_t}(s, a)] \\ &\quad - \pi_t(\tilde{\mathcal{A}}_t(s)|s) [\max_a A^{\pi_t}(s, a)] (\mathbb{E}_{a \sim \tilde{\pi}_t}[A^{\pi_t}(s, a)]) \\ &\geq \pi_t(\tilde{\mathcal{A}}_t(s)|s) [\max_a A^{\pi_t}(s, a)] \left[\max_a A^{\pi_t}(s, a) - \mathbb{E}_{a \sim \tilde{\pi}_t}[A^{\pi_t}(s, a)] \right] \\ &\geq \pi_t(\tilde{\mathcal{A}}_t(s)|s) [\max_a A^{\pi_t}(s, a)] \underbrace{\left[\max_a A^{\pi_t}(s, a) - \max_{a \notin \tilde{\mathcal{A}}_t(s)} A^{\pi_t}(s, a) \right]}_{:= \Delta^t(s)} \\ &= \pi_t(\tilde{\mathcal{A}}_t(s)|s) [\max_a A^{\pi_t}(s, a)] \Delta^t(s) \\ &\geq C_t \max_a A^{\pi_t}(s, a) \end{aligned}$$

□

Lemma 2. Using the update $\pi_{t+1}(a|s) = \pi_t(a|s) (1 + \eta A^{\pi_t}(s, a))$ with a step-size $\eta < \min \left\{ 1 - \gamma, \frac{1}{C_t(1-\gamma)} \right\}$, at any iteration t and state $s \in \mathcal{S}$, we have

$$V^*(s) - V^{\pi_{t+1}}(s) \leq [1 - \eta C_t(1 - \gamma)] [V^*(s) - V^{\pi_t}(s)]$$

where $C_t := \min_s \{ \pi_t(\tilde{\mathcal{A}}_t(s)|s) \Delta^t(s) \}$, $\tilde{\mathcal{A}}_t(s) := \arg \max_a Q^{\pi_t}(s, a)$, $\pi_t(\tilde{\mathcal{A}}_t(s)|s) = \sum_{a \in \tilde{\mathcal{A}}_t(s)} \pi_t(a|s)$, $\Delta^t(s) := \max_a Q^{\pi_t}(s, a) - \max_{a \notin \tilde{\mathcal{A}}_t(s)} Q^{\pi_t}(s, a)$, and $V^*(s)$ is the value function corresponding to the optimal policy π^* at $s \in \mathcal{S}$.

Proof. First, we use the value difference Lemma to show the SPMA update in Eq. (3) leads to a monotonic improvement in the value function.

$$V^{\pi_{t+1}}(s) - V^{\pi_t}(s) = \frac{1}{1 - \gamma} \mathbb{E}_{s \sim d^{\pi_{t+1}}} \left[\sum_a \pi_{t+1}(a|s) A^{\pi_t}(s, a) \right] \quad (12)$$

Plugging update Eq. (3) into the term within the brackets, we obtain the following:

$$\begin{aligned} \sum_a \pi_{t+1}(a|s) A^{\pi_t}(s, a) &= \sum_a \pi_t(a|s) A^{\pi_t}(s, a) [1 + \eta A^{\pi_t}(s, a)] \\ &= \sum_a \pi_t(a|s) A^{\pi_t}(s, a) + \eta \sum_a \pi_t(a|s) [A^{\pi_t}(s, a)]^2 \\ &= \eta \sum_a \pi_t(a|s) [A^{\pi_t}(s, a)]^2 \\ &> 0 \end{aligned}$$

Hence, $V^{\pi_{t+1}}(s) \geq V^{\pi_t}(s)$. Using Lemma 1, we have:

$$\eta \sum_a \pi_t(a|s) [A^{\pi_t}(s, a)]^2 \geq \eta C_t \max_a A^{\pi_t}(s, a) \quad (13)$$

Combining the above with the result from the value difference Lemma we have:

$$\begin{aligned} \sum_a \pi_{t+1}(a|s) A^{\pi_t}(s, a) &= \eta \sum_a \pi_t(a|s) [A^{\pi_t}(s, a)]^2 \\ &\geq \eta C_t \max_a A^{\pi_t}(s, a) \end{aligned} \quad (14)$$

We now show a linear convergence when using the update in Eq. (3). Let T be the Bellman optimality operator defined as:

$$(Tv)(s) = \max_a \{r(s, a) + \gamma \sum_{s'} \Pr[s'|s, a] v(s')\}$$

Applying the operator at iteration t we obtain:

$$TV^{\pi_t}(s) - V^{\pi_t}(s) = \max_a Q^{\pi_t}(s, a) - V^{\pi_t}(s) = \max_a A^{\pi_t}(s, a) \quad (15)$$

Let T^π be an operator w.r.t π defined as:

$$T^\pi(v) = \sum_a \pi(a|s) r(s, a) + \gamma \sum_a \pi(a|s) \sum_{s'} \Pr[s'|s, a] v(s')$$

Applying T^π to $V^{\pi'}(s)$ results in:

$$\begin{aligned} T^\pi V^{\pi'}(s) &= \sum_a \pi(a|s) r(s, a) + \gamma \sum_a \pi(a|s) \sum_{s'} \Pr[s'|s, a] V^{\pi'}(s) \\ &= \sum_a \pi(a|s) Q^{\pi'}(s, a) \end{aligned}$$

Using the above we obtain:

$$\begin{aligned} T^{\pi_{t+1}} V^{\pi_t}(s) - V^{\pi_t}(s) &= \sum_a \pi_{t+1}(a|s) A^{\pi_t}(s, a) \\ &\geq \eta C_t \max_a A^{\pi_t}(s, a) \quad (\text{using Ineq. 14}) \\ &= \eta C_t [TV^{\pi_t}(s) - V^{\pi_t}(s)] \quad (\text{using Eq. 15}) \end{aligned}$$

Assuming π^* is the optimal policy we have:

$$\begin{aligned} V^*(s) - V^{\pi_{t+1}}(s) &= V^*(s) - T^{\pi_{t+1}} V^{\pi_{t+1}}(s) && (\text{since } T^\pi V^\pi(s) = V^\pi(s)) \\ &\leq V^*(s) - T^{\pi_{t+1}} V^{\pi_t}(s) && (\text{since } V^{\pi_{t+1}}(s) \geq V^{\pi_t}(s) \ \forall s) \\ &= V^*(s) - V^{\pi_t}(s) - [T^{\pi_{t+1}} V^{\pi_t}(s) - V^{\pi_t}(s)] && (\text{add and subtract } V^{\pi_t}(s)) \\ &\leq V^*(s) - V^{\pi_t}(s) - \eta C_t [TV^{\pi_t}(s) - V^{\pi_t}(s)] \\ &= \eta C_t [V^*(s) - V^{\pi_t}(s)] + (1 - \eta C_t) [V^*(s) - V^{\pi_t}(s)] - \eta C_t [TV^{\pi_t}(s) - V^{\pi_t}(s)] \\ &= \eta C_t [TV^*(s) - V^{\pi_t}(s) - TV^{\pi_t}(s) + V^{\pi_t}(s)] + (1 - \eta C_t) [V^*(s) - V^{\pi_t}(s)] \\ &= \eta C_t [TV^*(s) - TV^{\pi_t}(s)] + (1 - \eta C_t) [V^*(s) - V^{\pi_t}(s)] \\ &\leq \gamma \eta C_t [V^*(s) - V^{\pi_t}(s)] + (1 - \eta C_t) [V^*(s) - V^{\pi_t}(s)] && (T \text{ is a } \gamma \text{ contraction map}) \\ &= [1 - \eta C_t(1 - \gamma)] [V^*(s) - V^{\pi_t}(s)] \end{aligned}$$

□

Theorem 2. Using the SPMA update in Eq. (3) with a step-size $\eta < \min \left\{ 1 - \gamma, \frac{1}{C_t(1-\gamma)} \right\}$ converges as:

$$\left\| V^{\pi^*} - V^{\pi_T} \right\|_{\infty} \leq \left(\prod_{t=0}^{T-1} \alpha_t \right) \left\| V^{\pi^*} - V^{\pi_0} \right\|_{\infty},$$

where $\alpha_t := (1 - \eta C_t(1 - \gamma))$, $C_t := \min_s \{ \pi_t(\tilde{a}_t(s)|s) \Delta^t(s) \}$, $\tilde{a}_t(s) := \arg \max_a Q^{\pi_t}(s, a)$ and $\Delta^t(s) := \max_a Q^{\pi_t}(s, a) - \max_{a \neq \tilde{a}} Q^{\pi_t}(s, a)$.

Proof. Using Lemma 2 we have

$$V^*(s) - V^{\pi_{t+1}}(s) \leq [1 - \eta C_t(1 - \gamma)] [V^*(s) - V^{\pi_t}(s)]$$

If $\eta < \frac{1}{C_t(1-\gamma)}$, both sides of the inequality above are positive leading to $|V^*(s) - V^{\pi_{t+1}}(s)| \leq (1 - \eta C_t(1 - \gamma)) |V^*(s) - V^{\pi_t}(s)|$. This is true for all $s \in \mathcal{S}$, hence we have:

$$\begin{aligned} \left\| V^{\pi^*} - V^{\pi_{t+1}} \right\|_{\infty} &\leq (1 - \eta C_t(1 - \gamma)) \left\| V^{\pi^*} - V^{\pi_t} \right\|_{\infty} \\ &= \alpha_t \left\| V^{\pi^*} - V^{\pi_t} \right\|_{\infty} \end{aligned}$$

Recurring from $t = 0$ to $t = T - 1$ we obtain a linear convergence:

$$\left\| V^{\pi^*} - V^{\pi_T} \right\|_{\infty} \leq \left(\prod_{t=0}^{T-1} \alpha_t \right) \left\| V^{\pi^*} - V^{\pi_0} \right\|_{\infty}$$

□

B.2 Function Approximation With Exact Advantage

Recall the definitions of $\tilde{\ell}_t$ and ℓ_t

$$\begin{aligned} \tilde{\ell}_t(\theta) &= \sum_s d^{\pi_t}(s) \text{KL}(\pi_{t+1/2}(\cdot|s) \parallel \pi_{\theta}(\cdot|s)) \\ \ell_t(\theta) &= \sum_{s \sim \tau} \text{KL}(h(f_{\theta_t}(s, \cdot))(1 + \eta A^{\pi_t}(s, \cdot)) \parallel h(f_{\theta}(s, \cdot))) \end{aligned}$$

Theorem 3. Under assumption 1-3, Algorithm 1 with $\eta < \min \left\{ 1 - \gamma, \frac{1}{C_t(1-\gamma)} \right\}$ converges as,

$$\begin{aligned} J(\pi^*) - J(\pi_T) \\ \leq \left(\prod_{t=0}^{T-1} \alpha_t \right) (J(\pi^*) - J(\pi_0)) + \beta \sum_{t=0}^{T-1} \prod_{i=t+1}^{T-1} \alpha_i, \end{aligned}$$

where $\beta = \frac{\sqrt{2}}{(1-\gamma)^2 \rho_{\min}} \sqrt{\epsilon_{\text{stat}} + \epsilon_{\text{bias}}}$ and α_t has the same definition as in Theorem 2.

Proof. We assumed that $z_{\theta}(s, a) := f_{\theta}(s, a) \forall (s, a)$ and $z_t(s, a) = f_{\theta_{t+1}}(s, a)$ where $f_{\theta} : \mathbb{R}^{SA} \rightarrow \mathbb{R}$ is a complex, non-linear model. We remind the following updates:

$$z_{t+1/2} = \arg \max_{z \in \mathbb{R}^{|S||A|}} \{ \langle \nabla_z J(z_t), z \rangle - 1/\eta D_{\Phi}(z, z_t) \}$$

$$\nabla \Phi(z_{t+1/2}) = \nabla \Phi(z_t) + \eta \nabla_z J(z_t)$$

(Mirror Ascent update without projection)

$$\pi_{t+1/2} = h(z_{t+1/2})$$

(h is softmax)

$$\pi_{t+1/2}(a|s) = \pi_t(a|s)(1 + \eta A^{\pi_t}(s, a))$$

$$\theta_{t+1} = (\text{S})\text{GD}(\ell_t(\theta))$$

(using (Stochastic)Gradient Descent for m iteration to minimize ℓ_t)

$$\pi_{t+1} = h(z_{t+1})$$

$z_{t+1/2}$ is an unprojected update for the tabular setting and therefore using Lemma 2 we have:

$$V^*(s) - V^{\pi_{t+1/2}}(s) \leq [1 - \eta C_t(1 - \gamma)] [V^*(s) - V^{\pi_t}(s)]$$

By adding and removing $V^{\pi_{t+1}}(s)$ to both sides and rearranging we have

$$V^*(s) - V^{\pi_{t+1}}(s) \leq [1 - \eta C_t(1 - \gamma)] [V^*(s) - V^{\pi_t}(s)] + V^{\pi_{t+1/2}}(s) - V^{\pi_{t+1}}(s).$$

Taking the expectation w.r.t. ρ , we obtain:

$$J(\pi^*) - J(\pi_{t+1}) \leq [1 - \eta C_t(1 - \gamma)] [J(\pi^*) - J(\pi_t)] + \underbrace{J(\pi_{t+1/2}) - J(\pi_{t+1})}_{:=E_1}$$

The term E_1 can be bounded as follows:

$$\begin{aligned} E_1 &= \sum_s d^{\pi_{t+1/2}}(s) \langle Q^{\pi_{t+1}}(s, \cdot), \pi_{t+1/2}(\cdot|s) - \pi_{t+1}(\cdot|s) \rangle \\ &\leq \sum_s d^{\pi_{t+1/2}}(s) \|Q^{\pi_{t+1}}(s, \cdot)\|_\infty \|\pi_{t+1/2}(\cdot|s) - \pi_{t+1}(\cdot|s)\|_1 && \text{(Holder inequality)} \\ &\leq \frac{1}{1 - \gamma} \sum_s d^{\pi_{t+1/2}}(s) \|\pi_{t+1/2}(\cdot|s) - \pi_{t+1}(\cdot|s)\|_1 && \text{(since } \|Q^{\pi_{t+1}}(s, \cdot)\|_\infty \leq \frac{1}{1 - \gamma}) \\ &= \frac{1}{1 - \gamma} \sum_s \frac{d^{\pi_{t+1/2}}(s)}{\rho(s)} \rho(s) \|\pi_{t+1/2}(\cdot|s) - \pi_{t+1}(\cdot|s)\|_1 \\ &\leq \frac{1}{1 - \gamma} \left\| \frac{d^{\pi_{t+1/2}}}{\rho} \right\|_\infty \sum_s \rho(s) \|\pi_{t+1/2}(\cdot|s) - \pi_{t+1}(\cdot|s)\|_1 \\ &\leq \frac{1}{(1 - \gamma)\rho_{\min}} \sum_s \rho(s) \|\pi_{t+1/2}(\cdot|s) - \pi_{t+1}(\cdot|s)\|_1 && \text{(since } d^{\pi_{t+1/2}}(s) \leq 1 \text{ and using assumption 3)} \\ &\leq \frac{1}{(1 - \gamma)^2 \rho_{\min}} \sum_s d^{\pi_t}(s) \|\pi_{t+1/2}(\cdot|s) - \pi_{t+1}(\cdot|s)\|_1 && \text{(since } d^{\pi_t} \geq (1 - \gamma)\rho) \\ &\leq \frac{\sqrt{2}}{(1 - \gamma)^2 \rho_{\min}} \sum_s d^{\pi_t}(s) \sqrt{\text{KL}(\pi_{t+1/2}(\cdot|s) \parallel \pi_{t+1}(\cdot|s))} \\ &\hspace{15em} \text{(using strong convexity of KL divergence or Pinsker's inequality)} \\ &\leq \frac{\sqrt{2}}{(1 - \gamma)^2 \rho_{\min}} \sqrt{\underbrace{\sum_s d^{\pi_t}(s) \text{KL}(\pi_{t+1/2}(\cdot|s) \parallel \pi_{t+1}(\cdot|s))}_{:=E_2}} && \text{(due to concavity of } \sqrt{\cdot} \text{ and Jensen's inequality)} \end{aligned}$$

where E_2 can be bounded as follows.

$$\begin{aligned} E_2 &= \tilde{\ell}_t(\theta_{t+1}) \\ &= \tilde{\ell}_t(\theta_{t+1}) - \min_{\theta} \tilde{\ell}_t(\theta_{t+1}) + \min_{\theta} \tilde{\ell}_t(\theta_{t+1}) \\ &\leq \epsilon_{\text{stat}} + \min_{\theta} \tilde{\ell}_t(\theta_{t+1}) && \text{(using assumption 1)} \\ &\leq \epsilon_{\text{stat}} + \epsilon_{\text{bias}} && \text{(using assumption 2)} \end{aligned}$$

Putting everything together we have:

$$E_1 \leq \underbrace{\frac{\sqrt{2}}{(1 - \gamma)^2 \rho_{\min}} \sqrt{\epsilon_{\text{stat}} + \epsilon_{\text{bias}}}}_{:=\beta}$$

Therefore we have

$$J(\pi^*) - J(\pi_{t+1}) \leq \underbrace{[1 - \eta C_t(1 - \gamma)]}_{\alpha_t} [J(\pi^*) - J(\pi_t)] + \beta$$

Unrolling the above recursion for T iterations,

$$J(\pi^*) - J(\pi_T) \leq \left(\prod_{t=0}^{T-1} \alpha_t \right) (J(\pi^*) - J(\pi_0)) + \beta \sum_{t=0}^{T-1} \prod_{i=t+1}^{T-1} \alpha_i$$

□

B.3 Function Approximation With Inexact Advantage

In practice computing the A^{π_t} at every iteration is costly. In this section, we assume that we access an oracle such that at each iteration t , it gives us \hat{A}^{π_t} an approximation of A^{π_t} .

Assumption 4. *Valid Approximation.* For all iterations t and $(s, a) \in \mathcal{S} \times \mathcal{A}$, $|\hat{A}^{\pi_t}(s, a)| \leq \frac{1}{1-\gamma}$.

Assumption 5. *Approximation Error.* For all iterations t and $s \in \mathcal{S}$, $\|A^{\pi_t}(s, \cdot) - \hat{A}^{\pi_t}(s, \cdot)\|_{\infty} \leq \epsilon_{approx}$.

Using this inexact advantage function, we define the following update and functions

$$\begin{aligned} \pi_{t+1/2}(a|s) &= \pi_t(a|s)(1 + \eta \hat{A}^{\pi_t}(s, a)) && \text{(replacing } A^{\pi_t} \text{ with } \hat{A}^{\pi_t}) \\ \tilde{\ell}_t(\theta) &= \sum_s d^{\pi_t}(s) \text{KL}(\pi_{t+1/2}(\cdot|s) \parallel \pi_{\theta}(\cdot|s)) \\ \ell_t(\theta) &= \sum_{s \sim \tau} \text{KL} \left(h(f_{\theta_t}(s, \cdot))(1 + \eta \hat{A}^{\pi_t}(s, \cdot)) \parallel h(f_{\theta}(s, \cdot)) \right) \end{aligned}$$

Since we use the inexact advantage, we cannot reuse the result of Lemma 2. So we provide a variant of that lemma with an inexact advantage.

Lemma 3. *Using the update $\pi_{t+1}(a|s) = \pi_t(a|s)(1 + \eta \hat{A}^{\pi_t}(s, a))$ with (i) a step-size $\eta < \min \left\{ 1 - \gamma, \frac{1}{C_t(1-\gamma)} \right\}$ and (ii) \hat{A}^{π_t} satisfying assumptions 4 and 5, at any iteration t and $s \in \mathcal{S}$ we have*

$$V^*(s) - V^{\pi}(s) \leq [1 - \eta C_t(1 - \gamma)] [V^*(s) - V^{\pi_t}(s)] + \frac{\epsilon_{approx}}{1 - \gamma}$$

where $C_t := \min_s \{\pi_t(\tilde{\mathcal{A}}_t(s)|s) \Delta^t(s)\}$, $\tilde{\mathcal{A}}_t(s) := \arg \max_a Q^{\pi_t}(s, a)$, $\pi_t(\tilde{\mathcal{A}}_t(s)|s) = \sum_{a \in \tilde{\mathcal{A}}_t(s)} \pi_t(a_t(s)|s)$ and $\Delta^t(s) := \max_a Q^{\pi_t}(s, a) - \max_{a \notin \tilde{\mathcal{A}}} Q^{\pi_t}(s, a)$, and $V^*(s)$ is the value function corresponding to the optimal policy π^* at $s \in \mathcal{S}$.

Proof. First, we use the value difference Lemma for a state $s \in \mathcal{S}$

$$\begin{aligned}
 V^\pi(s) - V^{\pi_t}(s) &= \frac{1}{1-\gamma} \mathbb{E}_{s' \sim d^\pi} \left[\sum_a \pi(a|s') A^{\pi_t}(s', a) \right] \\
 &= \frac{1}{1-\gamma} \mathbb{E}_{s' \sim d^\pi} \left[\sum_a \pi_t(a|s') (1 + \eta \hat{A}^{\pi_t}(s', a)) A^{\pi_t}(s', a) \right] \\
 &= \frac{1}{1-\gamma} \mathbb{E}_{s' \sim d^\pi} \left[\sum_a \eta \pi_t(a|s') \hat{A}^{\pi_t}(s', a) A^{\pi_t}(s', a) \right] \\
 &= \frac{1}{1-\gamma} \mathbb{E}_{s' \sim d^\pi} \left[\sum_a \eta \pi_t(a|s') (\hat{A}^{\pi_t}(s', a) - A^{\pi_t}(s', a) + A^{\pi_t}(s', a)) A^{\pi_t}(s', a) \right] \\
 &= \frac{1}{1-\gamma} \mathbb{E}_{s' \sim d^\pi} \left[\underbrace{\sum_a \eta \pi_t(a|s') (A^{\pi_t}(s', a))^2}_{:=T_1} \right] \\
 &\quad + \frac{1}{1-\gamma} \mathbb{E}_{s' \sim d^\pi} \left[\underbrace{\sum_a \eta \pi_t(a|s') (\hat{A}^{\pi_t}(s', a) - A^{\pi_t}(s', a)) A^{\pi_t}(s', a)}_{:=T_2} \right]
 \end{aligned}$$

T_1 can be bounded using Lemma 1,

$$T_1 \geq \eta C_t \max_a A^{\pi_t}(s, a)$$

To bound T_2 , we use *assumption 5*,

$$\begin{aligned}
 T_2 &\geq -\eta \left[\sum_a \pi_t(a|s') |(\hat{A}^{\pi_t}(s', a) - A^{\pi_t}(s', a))| |A^{\pi_t}(s', a)| \right] \\
 &\geq -\eta \left[\sum_a \pi_t(a|s') |(\hat{A}^{\pi_t}(s', a) - A^{\pi_t}(s', a))| \frac{1}{1-\gamma} \right] && (\text{since } A^\pi \leq 1/(1-\gamma)) \\
 &\geq -\eta \left[\sum_a \pi_t(a|s') \frac{\epsilon_{approx}}{1-\gamma} \right] && (\text{using assumption 5}) \\
 &= -\frac{\eta \epsilon_{approx}}{1-\gamma}
 \end{aligned}$$

Using the lower-bound for T_1 and T_2 we have

$$\sum_a \pi(a|s) A^{\pi_t}(s, a) \geq \eta C_t \max_a A^{\pi_t}(s, a) - \frac{\eta \epsilon_{approx}}{1-\gamma} \quad (16)$$

Putting everything together we have,

$$V^\pi(s) \geq V^{\pi_t}(s) - \frac{\eta \epsilon_{approx}}{(1-\gamma)^2} \quad (17)$$

$$\geq V^{\pi_t}(s) - \frac{\epsilon_{approx}}{(1-\gamma)} \quad (\text{since } \eta \leq 1-\gamma)$$

$$\implies V^{\pi_t}(s) - V^\pi(s) \leq \frac{\epsilon_{approx}}{(1-\gamma)} \quad (18)$$

$$(19)$$

Let T be the Bellman optimality operator defined as:

$$(Tv)(s) = \max_a \{r(s, a) + \gamma \sum_{s'} \Pr[s'|s, a] v(s')\}$$

Applying the operator at iteration t we obtain:

$$TV^{\pi_t}(s) - V^{\pi_t}(s) = \max_a Q^{\pi_t}(s, a) - V^{\pi_t}(s) = \max_a A^{\pi_t}(s, a) \quad (20)$$

Let T^π be an operator w.r.t π defined as:

$$T^\pi(v) = \sum_a \pi(a|s)r(s, a) + \gamma \sum_a \pi(a|s) \sum_{s'} \Pr[s'|s, a]v(s')$$

Applying T^π to $V^{\pi'}(s)$ results in:

$$\begin{aligned} T^\pi V^{\pi'}(s) &= \sum_a \pi(a|s)r(s, a) + \gamma \sum_a \pi(a|s) \sum_{s'} \Pr[s'|s, a]V^{\pi'}(s) \\ &= \sum_a \pi(a|s)Q^{\pi'}(s, a) \end{aligned}$$

Using the above we obtain:

$$\begin{aligned} T^\pi V^{\pi_t}(s) - V^{\pi_t}(s) &= \sum_a \pi(a|s)A^{\pi_t}(s, a) \\ &\geq \eta C_t \max_a A^{\pi_t}(s, a) - \frac{\eta \epsilon_{approx}}{1 - \gamma} \quad (\text{using Eq. (16)}) \\ &= \eta C_t [TV^{\pi_t}(s) - V^{\pi_t}(s)] - \frac{\eta \epsilon_{approx}}{1 - \gamma} \quad (\text{using Eq. 20}) \\ &\geq \eta C_t [TV^{\pi_t}(s) - V^{\pi_t}(s)] - \epsilon_{approx} \quad (\text{since } \eta \leq 1 - \gamma) \end{aligned}$$

Using Eq. (18) we have

$$T^\pi V^{\pi_t}(s) - T^\pi V^\pi(s) = \gamma \sum_a \pi(a|s) \sum_{s'} \Pr[s'|s, a](V^{\pi_t}(s') - V^\pi(s')) \quad (21)$$

$$\leq \gamma \sum_a \pi(a|s) \sum_{s'} \Pr[s'|s, a] \frac{\epsilon_{approx}}{1 - \gamma} \quad (\text{using Eq. (18)})$$

$$= \frac{\gamma \epsilon_{approx}}{1 - \gamma} \quad (22)$$

Assuming π^* is the optimal policy we have:

$$\begin{aligned} V^*(s) - V^\pi(s) &= V^*(s) - T^\pi V^\pi(s) \quad (\text{since } T^\pi V^\pi(s) = V^\pi(s)) \\ &= V^*(s) - T^\pi V^{\pi_t}(s) + T^\pi V^{\pi_t}(s) - T^\pi V^\pi(s) \\ &\leq V^*(s) - T^\pi V^{\pi_t}(s) + \frac{\gamma \epsilon_{approx}}{1 - \gamma} \quad (\text{using Eq. (22)}) \\ &= V^*(s) - V^{\pi_t}(s) - [T^\pi V^{\pi_t}(s) - V^{\pi_t}(s)] + \frac{\gamma \epsilon_{approx}}{1 - \gamma} \quad (\text{add and subtract } V^{\pi_t}(s)) \\ &\leq V^*(s) - V^{\pi_t}(s) - \eta C_t [TV^{\pi_t}(s) - V^{\pi_t}(s)] + \epsilon_{approx} + \frac{\gamma \epsilon_{approx}}{1 - \gamma} \\ &= \eta C_t [V^*(s) - V^{\pi_t}(s)] + (1 - \eta C_t) [V^*(s) - V^{\pi_t}(s)] - \eta C_t [TV^{\pi_t}(s) - V^{\pi_t}(s)] + \frac{\epsilon_{approx}}{1 - \gamma} \\ &= \eta C_t [TV^*(s) - V^{\pi_t}(s) - TV^{\pi_t}(s) + V^{\pi_t}(s)] + (1 - \eta C_t) [V^*(s) - V^{\pi_t}(s)] + \frac{\epsilon_{approx}}{1 - \gamma} \\ &= \eta C_t [TV^*(s) - TV^{\pi_t}(s)] + (1 - \eta C_t) [V^*(s) - V^{\pi_t}(s)] + \frac{\epsilon_{approx}}{1 - \gamma} \\ &\leq \gamma \eta C_t [V^*(s) - V^{\pi_t}(s)] + (1 - \eta C_t) [V^*(s) - V^{\pi_t}(s)] + \frac{\epsilon_{approx}}{1 - \gamma} \quad (T \text{ is a } \gamma \text{ contraction map}) \\ &= [1 - \eta C_t(1 - \gamma)] [V^*(s) - V^{\pi_t}(s)] + \frac{\epsilon_{approx}}{1 - \gamma} \end{aligned}$$

□

Theorem 5. Under assumption 1-5, Algorithm 1 with $\eta < \min \left\{ 1 - \gamma, \frac{1}{C_t(1-\gamma)} \right\}$ converges as,

$$\begin{aligned} & J(\pi^*) - J(\pi_T) \\ & \leq \left(\prod_{t=0}^{T-1} \alpha_t \right) (J(\pi^*) - J(\pi_0)) + \beta \sum_{t=0}^{T-1} \prod_{i=t+1}^{T-1} \alpha_i, \end{aligned}$$

where $\beta = \frac{\sqrt{2}}{(1-\gamma)^2 \rho_{\min}} \sqrt{\epsilon_{\text{stat}} + \epsilon_{\text{bias}}} + \frac{\epsilon_{\text{approx}}}{1-\gamma}$, $\alpha_t = [1 - \eta C_t(1-\gamma)]$, $C_t := \min_s \{ \pi_t(\tilde{\mathcal{A}}_t(s)|s) \Delta^t(s) \}$, $\tilde{\mathcal{A}}_t(s) := \arg \max_a Q^{\pi_t}(s, a)$, $\pi_t(\tilde{\mathcal{A}}_t(s)|s) = \sum_{a \in \tilde{\mathcal{A}}_t(s)} \pi_t(a_t(s)|s)$ and $\Delta^t(s) := \max_a Q^{\pi_t}(s, a) - \max_{a \notin \tilde{\mathcal{A}}} Q^{\pi_t}(s, a)$.

Proof. Using Lemma 3 with $\pi = \pi_{t+1/2}$,

$$V^*(s) - V^{\pi_{t+1/2}}(s) \leq [1 - \eta C_t(1-\gamma)] [V^*(s) - V^{\pi_t}(s)] + \frac{\epsilon_{\text{approx}}}{1-\gamma}$$

The rest of the proof is similar to the proof of Theorem 3. For completeness, we repeat it here. By adding and removing $V^{\pi_{t+1}}(s)$ to both sides and rearranging we have

$$V^*(s) - V^{\pi_{t+1}}(s) \leq [1 - \eta C_t(1-\gamma)] [V^*(s) - V^{\pi_t}(s)] + V^{\pi_{t+1/2}}(s) - V^{\pi_{t+1}}(s) + \frac{\epsilon_{\text{approx}}}{1-\gamma}.$$

Taking the expectation w.r.t. ρ we obtain:

$$J(\pi^*) - J(\pi_{t+1}) \leq [1 - \eta C_t(1-\gamma)] [J(\pi^*) - J(\pi_t)] + \underbrace{J(\pi_{t+1/2}) - J(\pi_{t+1})}_{:=E_1} + \frac{\epsilon_{\text{approx}}}{1-\gamma}.$$

The term E_1 can be bounded as follows:

$$\begin{aligned} E_1 &= \sum_s d^{\pi_{t+1/2}}(s) \langle Q^{\pi_{t+1}}(s, \cdot), \pi_{t+1/2}(\cdot|s) - \pi_{t+1}(\cdot|s) \rangle \\ &\leq \sum_s d^{\pi_{t+1/2}}(s) \|Q^{\pi_{t+1}}(s, \cdot)\|_\infty \|\pi_{t+1/2}(\cdot|s) - \pi_{t+1}(\cdot|s)\|_1 && \text{(Holder inequality)} \\ &\leq \frac{1}{1-\gamma} \sum_s d^{\pi_{t+1/2}}(s) \|\pi_{t+1/2}(\cdot|s) - \pi_{t+1}(\cdot|s)\|_1 && \text{(since } \|Q^{\pi_{t+1}}(s, \cdot)\|_\infty \leq \frac{1}{1-\gamma} \text{)} \\ &= \frac{1}{1-\gamma} \sum_s \frac{d^{\pi_{t+1/2}}(s)}{\rho(s)} \rho(s) \|\pi_{t+1/2}(\cdot|s) - \pi_{t+1}(\cdot|s)\|_1 \\ &\leq \frac{1}{1-\gamma} \left\| \frac{d^{\pi_{t+1/2}}}{\rho} \right\|_\infty \sum_s \rho(s) \|\pi_{t+1/2}(\cdot|s) - \pi_{t+1}(\cdot|s)\|_1 \\ &\leq \frac{1}{(1-\gamma)\rho_{\min}} \sum_s \rho(s) \|\pi_{t+1/2}(\cdot|s) - \pi_{t+1}(\cdot|s)\|_1 && \text{(since } d^{\pi_{t+1/2}}(s) \leq 1 \text{ and using assumption 3)} \\ &\leq \frac{1}{(1-\gamma)^2 \rho_{\min}} \sum_s d^{\pi_t}(s) \|\pi_{t+1/2}(\cdot|s) - \pi_{t+1}(\cdot|s)\|_1 && \text{(since } d^{\pi_t} \geq (1-\gamma)\rho \text{)} \\ &\leq \frac{\sqrt{2}}{(1-\gamma)^2 \rho_{\min}} \sum_s d^{\pi_t}(s) \sqrt{\text{KL}(\pi_{t+1/2}(\cdot|s) \parallel \pi_{t+1}(\cdot|s))} \\ &\hspace{15em} \text{(using strong convexity of KL divergence or Pinsker's inequality)} \\ &\leq \frac{\sqrt{2}}{(1-\gamma)^2 \rho_{\min}} \sqrt{\underbrace{\sum_s d^{\pi_t}(s) \text{KL}(\pi_{t+1/2}(\cdot|s) \parallel \pi_{t+1}(\cdot|s))}_{:=E_2}} && \text{(due to concavity of } \sqrt{\cdot} \text{)} \end{aligned}$$

where E_2 can be bounded as follows:

$$\begin{aligned}
E_2 &= \tilde{\ell}_t(\theta_{t+1}) \\
&= \tilde{\ell}_t(\theta_{t+1}) - \min_{\theta} \tilde{\ell}_t(\theta_{t+1}) + \min_{\theta} \tilde{\ell}_t(\theta_{t+1}) \\
&\leq \epsilon_{stat} + \min_{\theta} \tilde{\ell}_t(\theta_{t+1}) && \text{(using assumption 1)} \\
&\leq \epsilon_{stat} + \epsilon_{bias} && \text{(using assumption 2)}
\end{aligned}$$

Putting everything together we have:

$$E_1 \leq \underbrace{\frac{\sqrt{2}}{(1-\gamma)^2 \rho_{min}} \sqrt{\epsilon_{stat} + \epsilon_{bias}}}_{:=\beta'}$$

Therefore we have

$$J(\pi^*) - J(\pi_{t+1}) \leq \underbrace{[1 - \eta C_t(1 - \gamma)]}_{\alpha_t} [J(\pi^*) - J(\pi_t)] + \underbrace{\beta' + \frac{\epsilon_{approx}}{1 - \gamma}}_{:=\beta}.$$

Unrolling the above recursion for T iterations,

$$J(\pi^*) - J(\pi_T) \leq \left(\prod_{t=0}^{T-1} \alpha_t \right) (J(\pi^*) - J(\pi_0)) + \beta \sum_{t=0}^{T-1} \prod_{i=t+1}^{T-1} \alpha_i$$

□

C Tabular MDP Experiments

In this section, we empirically evaluate **SPMA** on tabular MDP environments. For these experiments, we use Cliff World (Sutton, 2018) and Frozen Lake (Brockman, 2016) following the setup in Vaswani et al. (2024). In subsection, C.1 we examine the case where the policy is parametrized using softmax tabular representation. In subsection, C.2 we investigate the function approximation setting as described in Section 4, where the policy is parametrized using a linear model.

C.1 Softmax Tabular Representation

For this parametrization we initialize $z \in \mathbb{R}^{S \times A}$ uniformly, i.e., $\pi_0(a|s) = \frac{1}{|A|}$ for each a and s . Furthermore, for each algorithm, we set η using a grid search and pick the step-sizes that result in the best area under the curve (AUC). The tabular MDP results suggest **SPMA** and **NPG** achieve similar performance and they both outperform **SPG** (Sutton et al., 1999; Schulman et al., 2017) (see Fig. 3). To analyze the sensitivity of each algorithm to the choice of η , we examine each optimizer across different values of η . The results in Fig. 4 suggest that overall **SPG** (in green) is more sensitive to different values of η compared to **SPMA** (blue) and **NPG** (red).

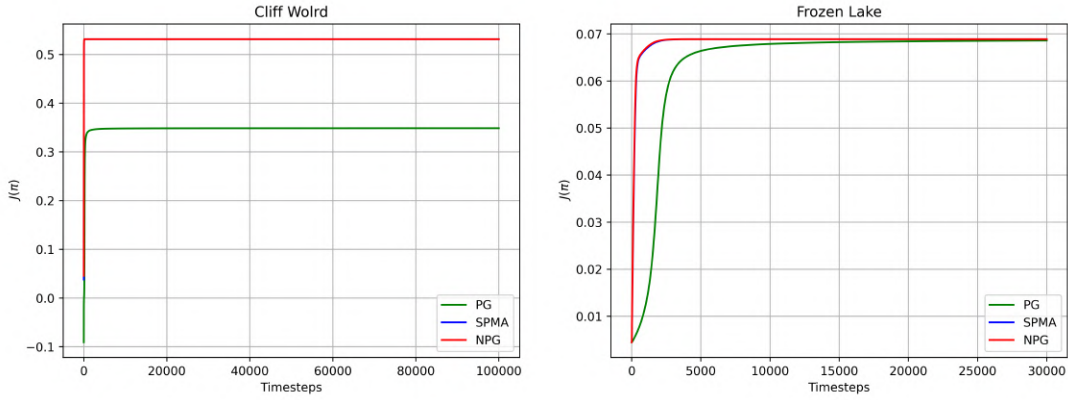


Figure 3: **SPMA** matches the performance of **NPG** and they both outperform **SPG**.

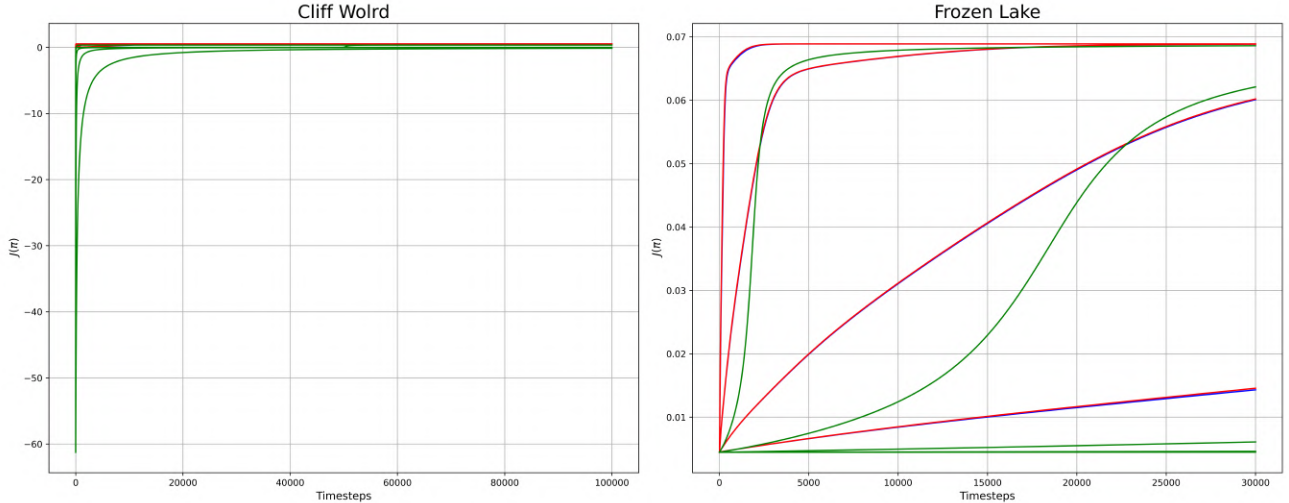


Figure 4: **SPG** (green) is more sensitive to η compared to **SPMA** and **NPG** (blue and red).

C.2 Linear Functional Approximation (Linear FA)

For the Linear FA setting, we use a log-linear policy parametrization: $\pi_t(a|s) = \frac{\exp(\mathbf{X}(s,a)\theta)}{\sum_{a'} \exp(\mathbf{X}(s,a')\theta)}$, with $\mathbf{X} \in \mathbb{R}^{S \times A \times d}$ and $\theta \in \mathbb{R}^d$ representing the features and parameters. We use constant initialization for θ and following Vaswani et al. (2024), use tile-coded features for \mathbf{X} . As in the previous section, we set η for SPMA and MDPO via grid search and report results based on the best AUC. For the inner loop optimization (e.g., minimizing Eq. (5)), we use Armijo line search (Armijo, 1966), avoiding an additional grid search for the step-size. For SPG we use the update from Mei et al. (2020) where Armijo line search is used to set η .

We make the following observations from the results: (i) SPG performs poorly in the linear FA setting, while both SPMA and MDPO perform well when the parameter dimension d and the number of inner loop optimizations m are sufficiently large. (ii) In the CW environment, for smaller d , SPMA converges faster than MDPO (Fig. 5, top row). Increasing m from 25 to 50 narrows the gap between SPMA and MDPO (top vs. bottom row). (iii) In the FL environment, SPMA and MDPO perform similarly, both outperforming SPG (Fig. 6).

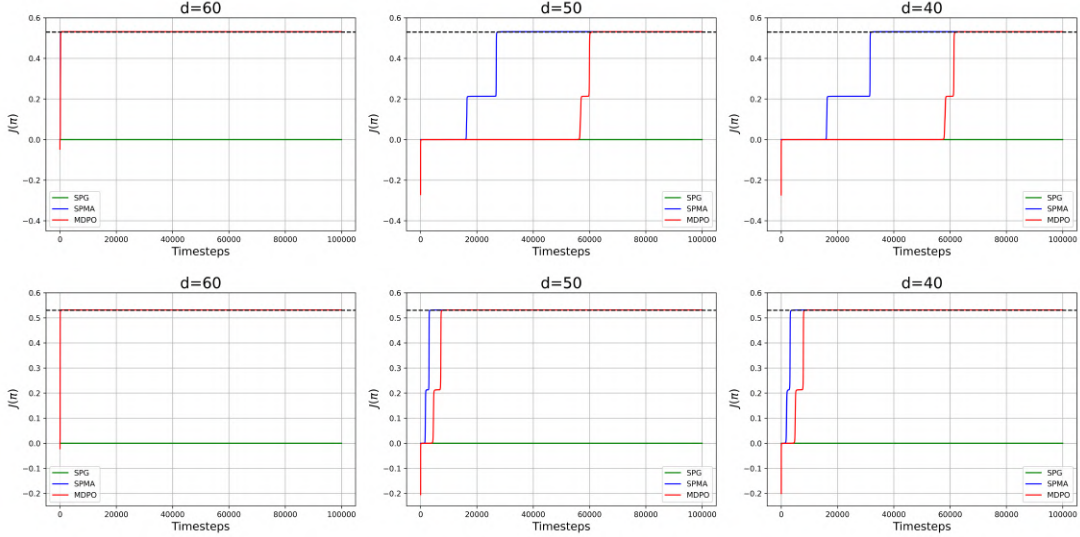


Figure 5: CW environment: The top row ($m = 25$) shows that SPMA converges faster than MDPO as d decreases, while the bottom row ($m = 50$) shows the gap decreases as the number of inner loop optimizations increases.

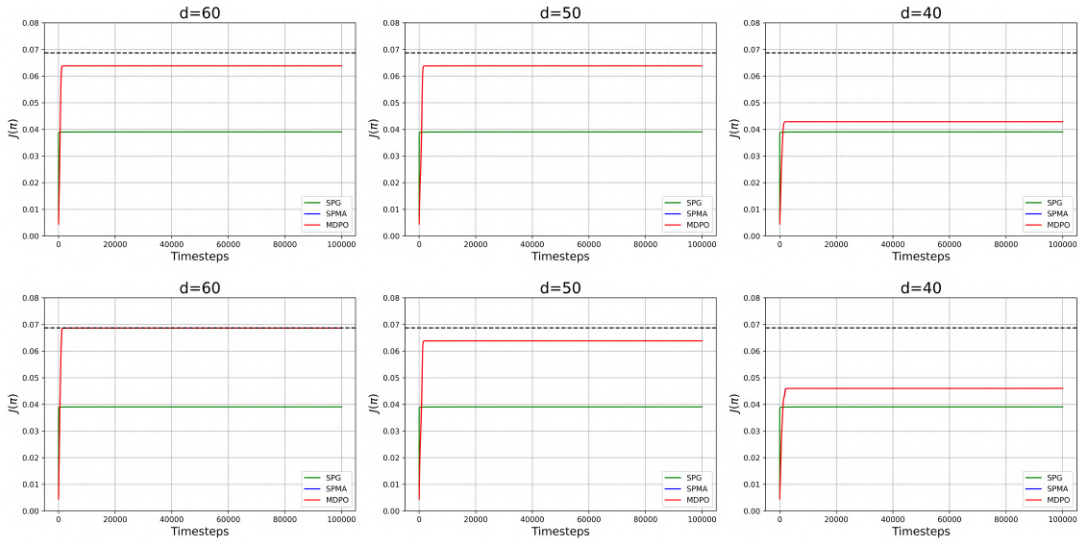


Figure 6: FL environment: The top row ($m = 25$) and bottom row ($m = 50$) show that SPMA and MDPO have similar convergence and both outperform SPG. The performance of both SPMA and MDPO improves as d increases (i.e., the bias decreases) and m increases (i.e., the optimization error decreases).

C.3 Empirical Verification that C_t is Non-zero

In this section, we empirically investigate the value of C_t . Recall that $C_t := \min_s \{\pi_t(\tilde{\mathcal{A}}_t(s)|s) \Delta^t(s)\}$, $\tilde{\mathcal{A}}_t(s) := \arg \max_a Q^{\pi_t}(s, a)$, $\pi_t(\tilde{\mathcal{A}}_t(s)|s) = \sum_{a \in \tilde{\mathcal{A}}_t(s)} \pi_t(a|s)$, and $\Delta^t(s) := \max_a Q^{\pi_t}(s, a) - \max_{a \notin \tilde{\mathcal{A}}_t(s)} Q^{\pi_t}(s, a)$. To this end, we compute the per-state metric $C_t(s) := \pi_t(\tilde{\mathcal{A}}_t(s)|s) \Delta^t(s)$ and its components, $\pi_t(\tilde{\mathcal{A}}_t(s)|s)$ and $\Delta^t(s)$, for the CliffWorld and FrozenLake environments using softmax tabular policy parametrization. Our results indicate that $C_t(s)$ is nonzero for all states except for certain terminal states, specifically the goal state and the hole states in FrozenLake, where $Q^{\pi}(s, a)$ is zero for all actions a and policies π . In these cases, we observe that $\Delta^t(s)$ becomes zero, leading to $C_t(s) = 0$ and consequently $C_t = 0$. Note that in the CliffWorld environment, the transition probability matrix moves the agent back to the starting point at terminal states, so they do not pose an issue. However, in the FrozenLake environment, the agent remains at the terminal state with probability one.

Given these empirical results, one could modify the theorems in this paper to exclude the problematic terminal states. For example, in Theorem 2, instead of bounding $V^{\pi^*}(s) - V^{\pi_T}(s)$ for all states using the ℓ_∞ norm, we could exclude states where $Q^{\pi^*}(s, a) = 0$ for all actions a . This exclusion is analogous to the bandit literature, which considers only arms with a nonzero gap in the regret bound. In Figure 7, we plot C_t using softmax tabular policy parameterizations for both CliffWorld and FrozenLake environments, confirming that after excluding terminal states, C_t is lower-bounded by a positive constant.

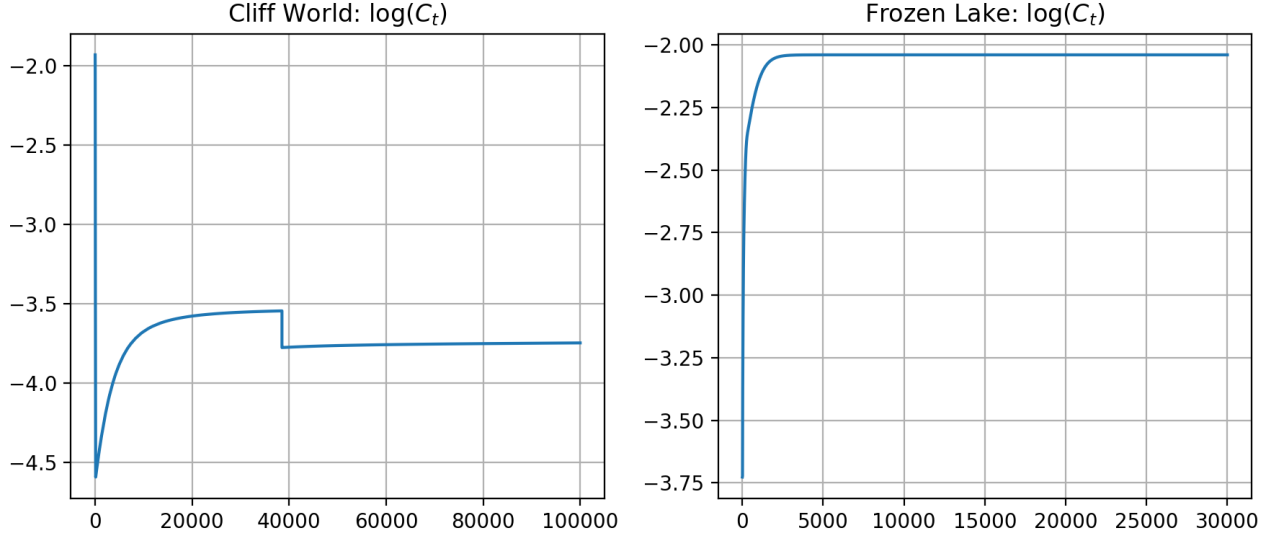


Figure 7: SPMA with softmax tabular policy parametrization: After excluding terminal states, results confirm that C_t is lower-bounded by a positive constant.

D Additional Details for Stable Baselines Experiments

In subsection D.1, we provide additional details on the hyper-parameters used for the results in Section 5. Next, we present an ablation study on the number of inner loop optimization steps (m) in subsection D.2.

D.1 Atari and Mujoco Details

In the Atari experiments, we use the default hyper-parameters for each method from stable baselines (Raffin et al., 2021). This choice is motivated by two factors: i) following the work of Tomar et al. (2020), we aim to evaluate the effectiveness of different surrogate losses without conducting an exhaustive search over numerous hyper-parameters; ii) the CNN-based actor and critic networks make grid searching over many hyper-parameters (e.g., framestack, GAE λ , horizon length, discount factor) computationally infeasible. For a complete list of hyper-parameters used in the Atari experiments, see Table 1.

In the MuJoCo experiments, we use the default hyper-parameters from stable baselines for each method, but perform a grid search on the Adam inner loop step size for PPO and TRPO-constrained (best among $[3 \times 10^{-3}, 3 \times 10^{-4}, 3 \times 10^{-5}]$) and the probability ratio clipping parameter in PPO (best from $[0.1, 0.2, 0.3]$). For the regularized surrogates (i.e., the remaining methods: SPMA, MDPO, and TRPO-regularized), we avoid a grid search for the inner loop step size by using a full batch (i.e., the horizon length) along with the Armijo line search (Armijo, 1966). See Table 2 for the full list of hyper-parameters used in the MuJoCo experiments.

To set η for the regularized surrogates, we perform a grid search over five fixed values ($[0.3, 0.5, 0.7, 0.9, 1.0]$). Although Tomar et al. (2020) anneals η from 1 to 0 during training, we observe that using a constant step size results in better performance. Our grid search strategy for all stable baselines experiments is consistent: we run the experiments for 2 million iterations, select the hyper-parameters that yield the best AUC, and then use these hyper-parameters for an additional 8 million iterations.

Hyperparameter	SPMA	MDPO	TRPO_regularized	TRPO_constrained	PPO
Reward normalization	✗	✗	✗	✗	✗
Observation normalization	✗	✗	✗	✗	✗
Orthogonal weight initialization	✓	✓	✓	✓	✓
Value function clipping	✗	✗	✗	✗	✗
Gradient clipping	✗	✗	✗	✗	✓
Probability ratio clipping	✗	✗	✗	✗	✓
Adam step-size	3×10^{-4}				
Minibatch size	256				
Framestack	4				
Number of environment copies	8				
GAE λ	0.95				
Horizon (T)	128				
Number of inner loop updates (m)	5				
Entropy coefficient	0				
Discount factor	0.99				
Total number of timesteps	10^7				
Number of runs for plot averages	5				
Confidence interval for plot runs	$\sim 95\%$				

Table 1: Hyper-parameters for Atari experiments.

Hyperparameter	SPMA	MDPO	TRPO_regularized	TRPO_constrained	PPO
Minibatch size	2048	2048	2048	64	64
Reward normalization	✗	✗	✗	✗	✗
Observation normalization	✗	✗	✗	✗	✗
Orthogonal weight initialization	✓	✓	✓	✓	✓
Value function clipping	✗	✗	✗	✗	✗
Gradient clipping	✗	✗	✗	✗	✓
Probability ratio clipping	✗	✗	✗	✗	✓
Adam step-size	✗	✗	✗	✓	✓
GAE λ	0.95				
Horizon (T)	2048				
Number of inner loop updates (m)	5				
Entropy coefficient	0				
Discount factor	0.99				
Total number of timesteps	10^7				
Number of runs for plot averages	5				
Confidence interval for plot runs	$\sim 95\%$				

Table 2: Hyper-parameters for MuJoCo experiments.

D.2 Ablation Study on the Number of Inner Loop Optimization Steps

In this subsection, we investigate the effect of varying the number of inner loop optimization steps (m) in the stable baselines experiments. Consistent with [Tomar et al. \(2020\)](#), we observe that using $m = 1$ results in poor performance, so we focus on larger values of m . In the MuJoCo experiments, increasing m from 5 to 10 and 15 consistently improves the performance of SPMA (see Figure 10). Specifically, for larger m , SPMA becomes comparable to TRPO-constrained on Hopper and Ant, while outperforming it on HalfCheetah (see Figure 8).

For the Atari experiments, we observe that increasing m does not necessarily improve the results across methods (see Figure 11). We conjecture that this is a side-effect of using a constant tuned step-size (for $m = 5$) in the inner-loop. In the future, we plan to run the full grid-search for the inner-loop step-size for each value of m . Alternatively, we plan to investigate an adaptive way of setting the inner-loop step-size.

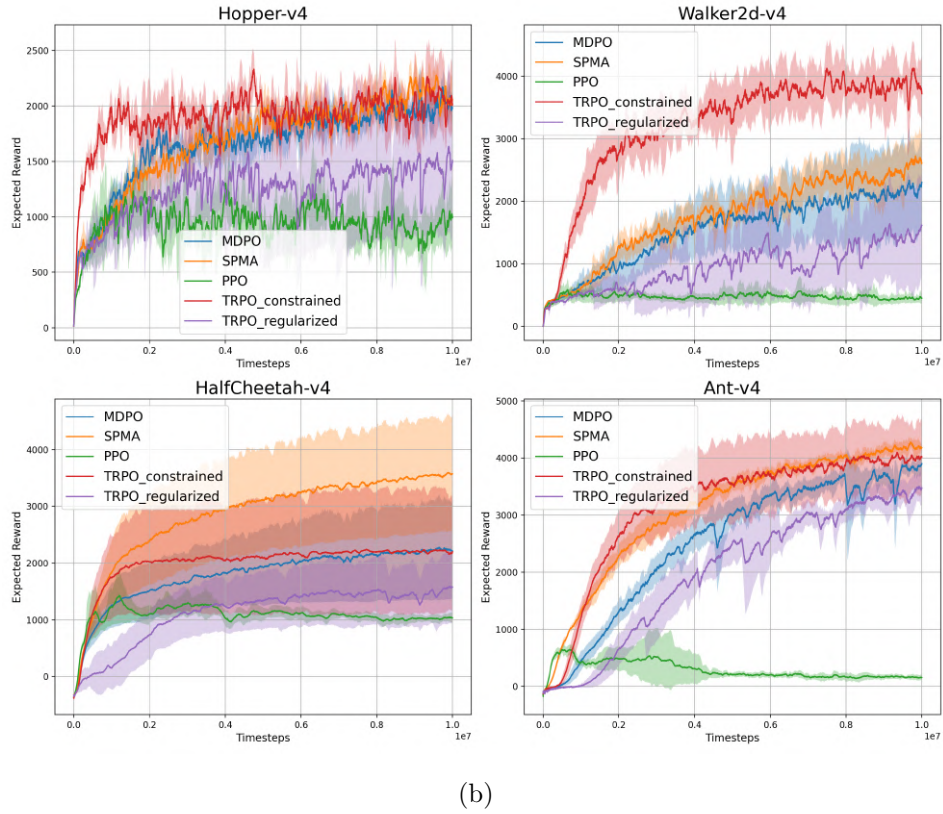
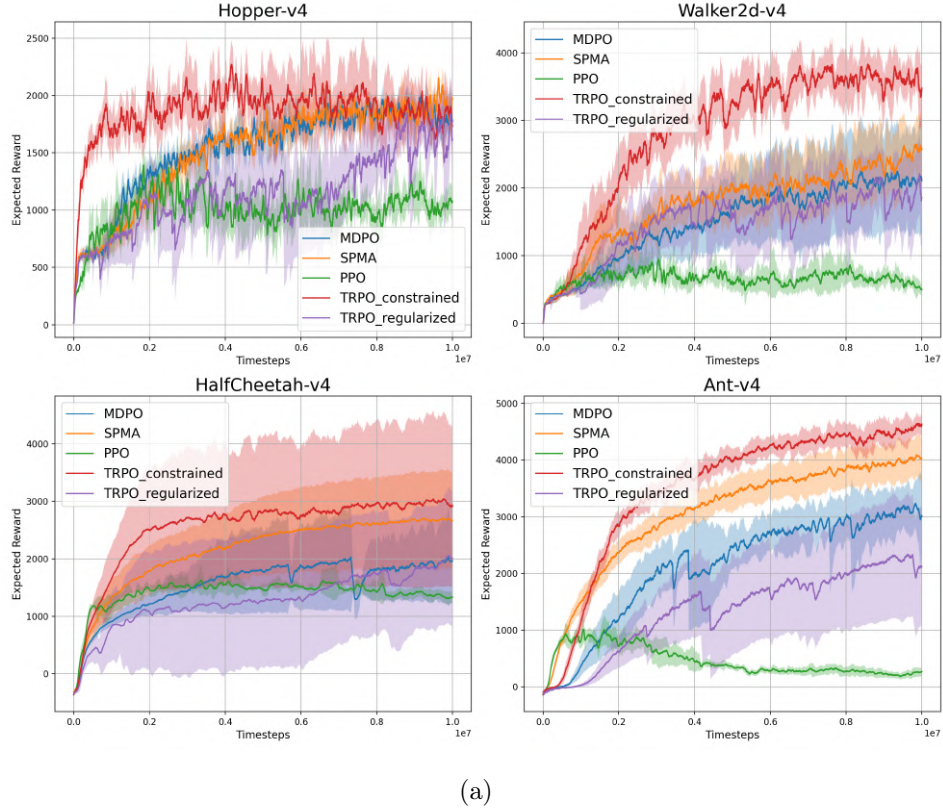


Figure 8: MuJoCo results for $m = 10$ (a) and $m = 15$ (b). As m increases from 5 (Figure 2) to 10 and 15, SPMA shows performance comparable to the fine-tuned TRPO-constrained.

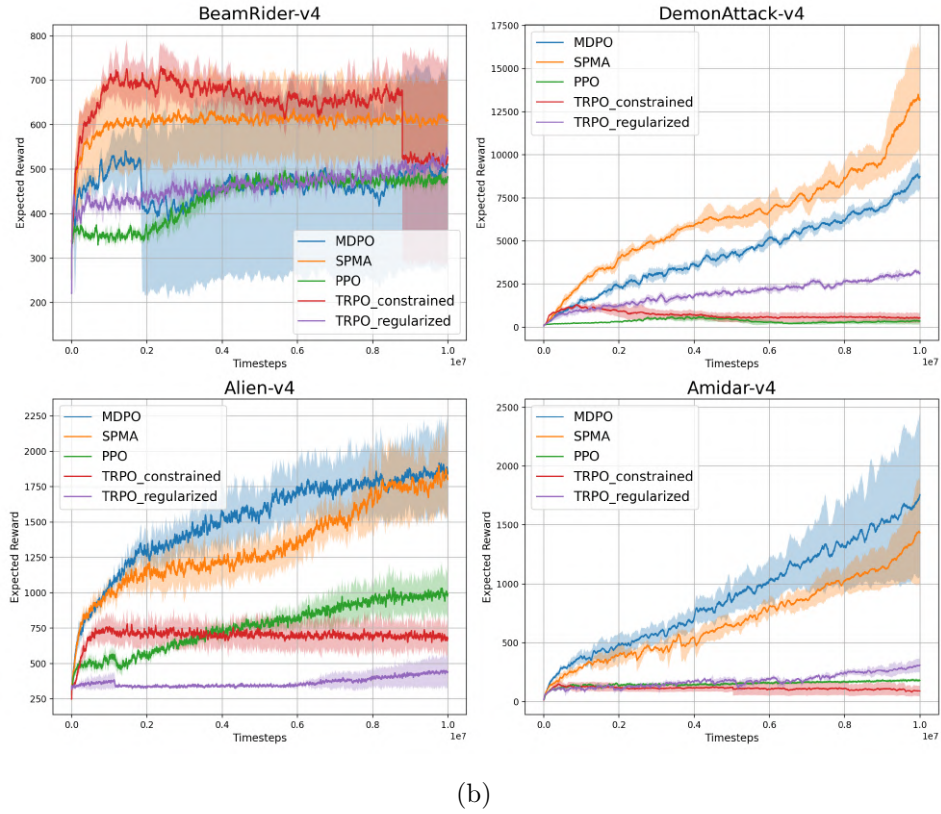
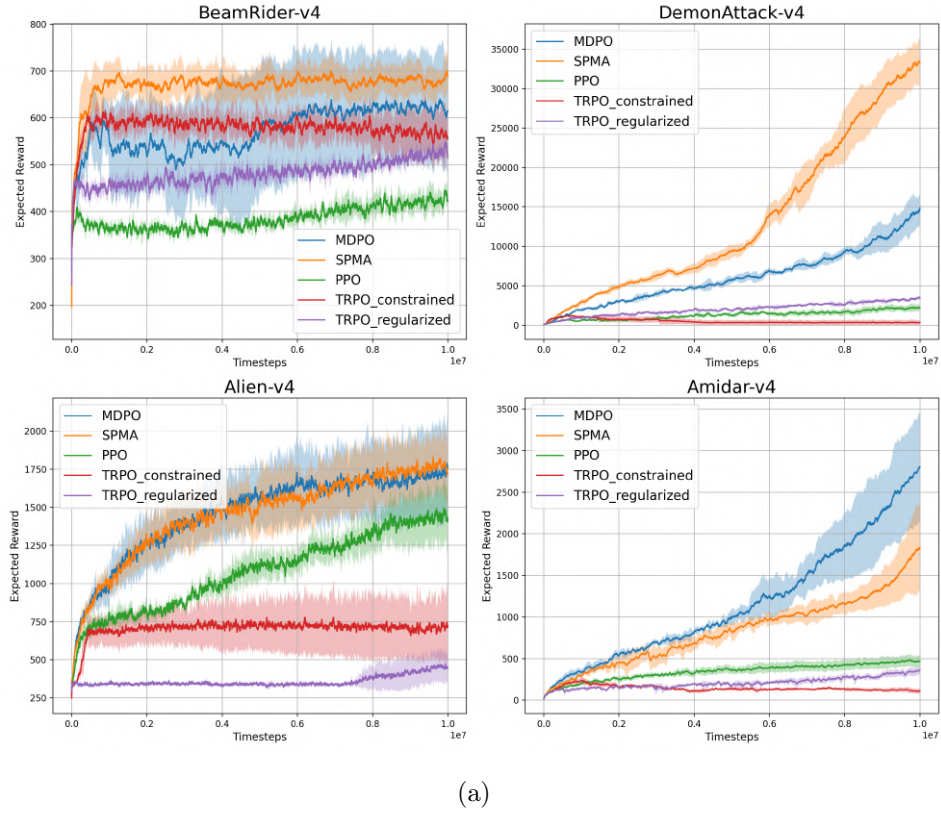


Figure 9: Atari results for $m = 10$ (top) and $m = 15$ (bottom). Increasing m does not necessarily lead to performance improvements.

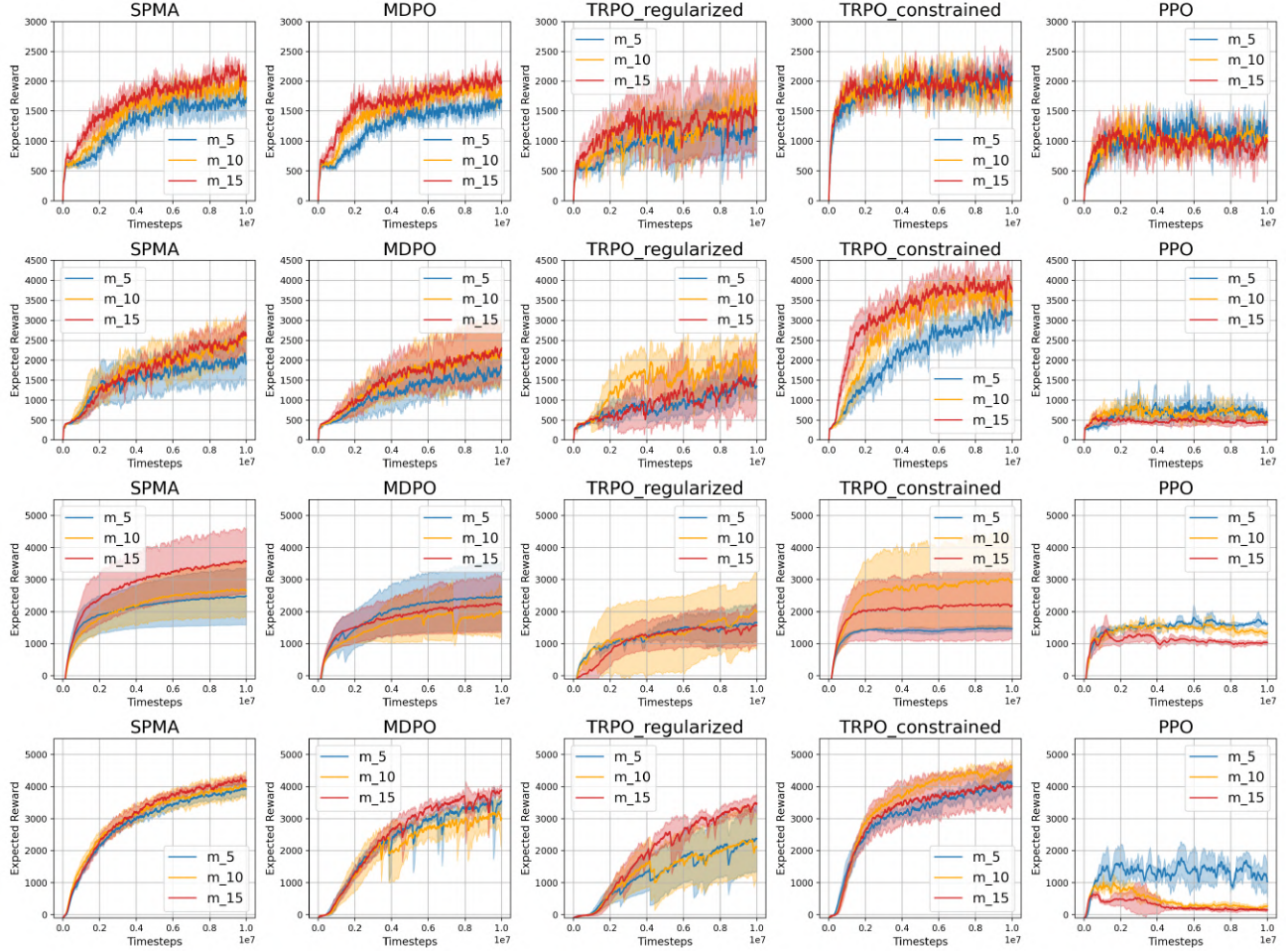


Figure 10: MuJoCo ablation on m : The rows correspond to the Hopper-v4, Walker2d-v4, HalfCheetah-v4, and Ant-v4 environments, respectively. As the number of inner loop optimization steps m increases, SPMA shows improvements in expected reward and becomes comparable to the fine-tuned TRPO-constrained.

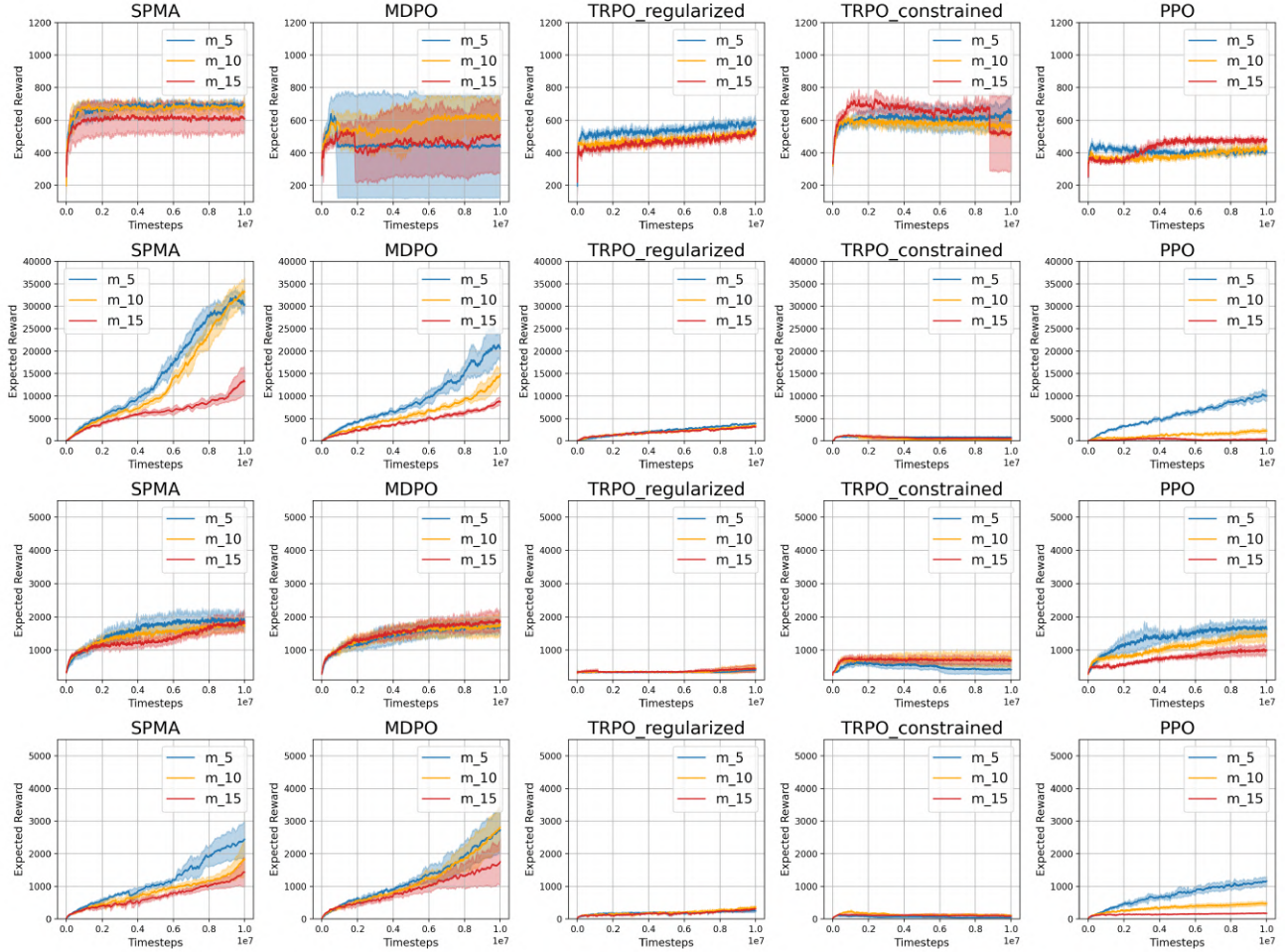


Figure 11: Atari ablation on m : The rows correspond to the BeamRider-v4, DemonAttack-v4, Alien-v4, and Amidar-v4 games. We observe that increasing m does not necessarily improve results across methods.