Fast Convergence of Softmax Policy Mirror Ascent for Bandits & Tabular MDPs

Reza Asad REZA_ASAD@SFU.CA
Simon Fraser University
Reza Babanezhad BABANEZHAD@GMAIL.COM
Samsung AI, Montréal
Issam Laradji ISSAM.LARADJI@GMAIL.COM
ServiceNow Research
Nicolas Le Roux NICOLAS@LE-ROUX.NAME
Mila, Université de Montréal, McGill
Sharan Vaswani VASWANI.SHARAN@GMAIL.COM
Simon Fraser University

Abstract

We analyze the convergence of a novel policy gradient algorithm (referred to as SPMA) for multi-armed bandits and tabular Markov decision processes (MDPs). SPMA is an instantiation of mirror ascent and uses the softmax parameterization with a log-sum-exp mirror map. Given access to the exact policy gradients, we prove that SPMA with a constant step-size requires $O(\log(1/\epsilon))$ iterations to converge to an ϵ-optimal policy. The resulting convergence rate is better than both the $\Theta\left({}^{1}/_{\epsilon}\right)$ rate for constant step-size softmax policy gradient (SPG) and the $O\left({}^{1}/\sqrt{\epsilon}\right)$ rate for SPG with Nesterov acceleration. Furthermore, unlike the SPG results, the convergence rate for SPMA does not depend on potentially large problem-dependent constants, and matches the rate achieved by natural policy gradient (NPG). Furthermore, for multi-armed bandits, we prove that SPMA with gap-dependent step-sizes can result in global super-linear convergence. Our experimental evaluations on tabular MDPs and continuous control tasks demonstrate that SPMA consistently outperforms SPG while achieving similar or better performance compared to NPG.

1. Introduction

Policy gradient (PG) methods have been critical to the achievements of deep reinforcement learning (RL) [20, 23]. While the PG objective is non-concave and thus potentially difficult to analyze, recent theoretical research [1, 3, 11, 13–16, 21] studied PG methods in simplified settings, exploiting the objective's properties to guarantee global convergence to an optimal policy. Such analyses of PG methods are helpful in understanding the underlying optimization issues and providing a systematic approach to designing new practical PG methods.

We focus on PG methods that parameterize the policy using the softmax function and consider the tabular setting where the number of parameters scales with the number of states and actions. Given access to the exact policy gradients (corresponding to the case where the rewards and transition probabilities are known or can be efficiently estimated), Agarwal et al. [1] proved that the common softmax policy gradient (SPG) method can attain asymptotic convergence to an optimal policy despite the non-concave nature of the PG objective. Mei et al. [15] improved this result and quantified the rate of convergence, proving that constant step-size SPG requires $\Theta\left({}^{1}/_{\epsilon}\right)$ iterations to converge to an ϵ-optimal policy. Recently, Chen et al. [6] combined constant step-size SPG with Nesterov acceleration to achieve an $O\left(1/\sqrt{\epsilon}\right)$ convergence; while [13, 14] use SPG with adaptive step-sizes to achieve a linear $O(\log(1/\epsilon))$ convergence rate. However, it is known that SPG incurs a dependence on the distribution mismatch ratio which can be exponential in the size of the state space [12], consequently rendering these rates vacuous.

On the other hand, policy mirror descent (PMD) [8, 11, 28] or natural policy gradient (NPG) [9] use a different policy parameterization. These methods directly parameterize the probability of taking an action in a particular state. In the tabular setting with access to exact policy gradients, these methods correspond to mirror ascent (with the negative entropy mirror map) in the space of probabilities and have been shown to achieve linear convergence to the optimal policy [3, 8, 11, 13, 28]. Furthermore, unlike SPG, NPG does not incur the bad dependence on the mismatch ratio.

Given these results, it is unclear whether PG methods using the softmax parameterization can lead to comparable results as NPG or PMD. In this paper, we resolve this question in the affirmative. In particular, we analyze a recently proposed algorithm [26] that uses the softmax parameterization and consists of a mirror ascent update with the log-sum-exp mirror map. We prove that the resulting update referred to as **S**oftmax **P**olicy **M**irror **A**scent (SPMA) results in convergence comparable to that of constant step-size NPG. In particular, we make the following contributions.

Contribution 1: Global linear convergence of SPMA: While Vaswani et al. [26] argue that SPMA has desirable theoretical properties when used with general function approximation, they only prove that it results in an $O\left(1/\epsilon\right)$ convergence to a stationary point. In contrast, for both the multi-armed bandit and tabular MDP settings, we prove that SPMA with a constant step-size and access to exact policy gradients achieves a linear $O(\log\left(1/\epsilon\right))$ convergence rate without a dependence on the distribution mismatch ratio, thus matching the convergence results of NPG (Section 4.1).

Contribution 2: Global super-linear convergence of SPMA for multi-armed bandits: For multi-armed bandits, we prove that SPMA with specific gap-dependent step-sizes can achieve a global super-linear convergence rate (Section 4.2).

Contribution 3: Experimental results: In Section 5, we empirically show that SPMA is more robust than SPG and matches the performance of NPG on tabular MDPs. In Appendix D, we follow the idea in Vaswani et al. [26] and describe how to effectively use the SPMA algorithm in more realistic settings that require handling stochasticity and function approximation. We use the resulting technique for continuous control Mujoco tasks and demonstrate that SPMA outperforms SPG, while resulting in similar or better performance compared to MDPO [25], a generalization of NPG.

2. Problem Formulation

We consider an infinite-horizon discounted Markov decision process (MDP) [17] 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, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta_{\mathcal{S}}$ is the transition probability function, $r : \mathcal{S} \times \mathcal{A} \to [0, 1]$ 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 only consider the tabular setting where the number of states and actions is finite. Given $s \in \mathcal{S}$, the policy π induces a probability distribution $\pi(.|s)$ over the actions. The action-value function $Q^{\pi} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ induced by π is $Q^{\pi}(s, a) := \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s, a_0 = a]$ where $s_t \sim p(.|s_{t-1}, a_{t-1})$, and $a_t \sim \pi(.|s)$ for $t \geq 1$. The value function corresponding to Q^{π} starting from state s is defined as $J_s(\pi) = \mathbb{E}_{a \sim \pi(.|s)}[Q^{\pi}(s, a)]$ with $J(\pi) := \mathbb{E}_{s \sim \rho}[J_s(\pi)]$ representing the expected discounted cumulative reward. Furthermore, the advantage function $A^{\pi} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is defined as $A^{\pi}(s, a) := Q^{\pi}(s, a) - J_s(\pi)$. The policy also induces a discounted state-occupancy measure $d^{\pi}(s) := (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t \Pr^{\pi}[s_t = s | s_0 \sim \rho]$.

Given this setup, the goal is to find the optimal policy π^* that maximizes the expected reward objective $J(\pi)$. 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. Throughout this paper, we parameterize the policy using a softmax representation, i.e., $\pi(a|s) = \frac{\exp(z(s,a))}{\sum_{a'} \exp(z(s,a'))}$, where $z \in \mathbb{R}^{\mathcal{S} \times \mathcal{A}}$. In the tabular setting, we aim to learn the SA-dimensional vector of logits. In the next section, we present `SPMA` that uses the softmax parameterization with the log-sum-exp mirror map.

3. Algorithm

Mirror ascent is an iterative algorithm whose update at iteration $t \in \{0, 1, \ldots, T-1\}$ is given as:

$$z_{t+1} = \arg\max \left[\langle z, \nabla_z J(z_t) \rangle - {}^1\!/\!_\eta\, D_\Phi(z, z_t)\right] \;\text{ or }\; \nabla\Phi(z_{t+1}) = \nabla\Phi(z_t) + \eta \nabla J(z_t) \tag{1}$$

where z_t is the logit at iteration t, η is the step-size, and D_Φ is the Bregman divergence (induced by the mirror map Φ) between the logits z and z_t. Using the policy gradient theorem [23], $[\nabla J(z_t)](s,a) = d^{\pi_t}(s)\, \pi_t(a|s)\, A^{\pi_t}(s,a)$. Following Vaswani et al. [26], we use the log-sum-exp mirror map: $\phi(z(s,.)) = \ln(\sum_a \exp(z(s,a))$ for each state s, implying that $D_\phi(z(s,\cdot), z_t(s,\cdot)) = \text{KL}(\pi_t(\cdot|s) || \pi(\cdot|s))$ [27, Lemma 11]. Weighting these per-state divergences by $d^\pi(s)$ results in the following update rule for each state $s \in \mathcal{S}$:

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

We first ensure that $\pi_{t+1}(\cdot|s)$ is a valid probability distribution. Assuming $r(s,a) \in [0,1]$, $\forall s \in \mathcal{S}$, $a \in \mathcal{A}$, $|A(s,a)| \leq \frac{1}{1-\gamma}$ implying that $\eta \leq 1 - \gamma$ is sufficient to guarantee that $\pi_{t+1}(a|s)$ is non-negative for every (s,a). We also ensure that $\sum_a \pi_{t+1}(s,a) = 1$ for each $s \in \mathcal{S}$: since $\sum_a \pi_t(a|s) A^{\pi_t}(s,a) = 0$, $\sum_a \pi_{t+1}(a|s) = \sum_a \pi_t(a|s) + \eta \sum_a \pi_t(a|s) A^{\pi_t}(s,a)) = 1$. For the bandit setting where $|\mathcal{S}| = 1$, the above update simplifies to:

$$\pi_{t+1}(a) = \pi_t(a)\, (1 + \eta\, [r(a) - \langle \pi_t, r \rangle]) = \pi_t(a)\, \left[1 + \eta \sum_{a' \neq a} \pi_t(a')\, \Delta(a, a')\right], \tag{3}$$

where $\Delta(a, a') := r(a) - r(a')$ represents the reward gap between arms a and a'. Similar to the MDP case, using $\eta < 1$ ensures that π_{t+1} is a valid probability distribution.

In contrast to the natural policy gradient (`NPG`) update: $\pi_{t+1}(a|s) \propto \pi_t(a|s)\, \exp(\eta\, A^{\pi_t}(s,a))$, the `SPMA` update in Eq. (2) 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. The softmax policy gradient (`SPG`) update 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 subsequently, this desirable property enables `SPMA` to achieve faster rates than `SPG`. Finally, we note that for the bandit case, the update rule in Eq. (3) is related to the `PROD` algorithm [5] used in the online learning literature. In contrast to `SPMA` which is derived from mirror ascent, the `PROD` algorithm is derived using a linearization of the Hedge [7] algorithm and requires explicit normalization to obtain probabilities.

4. Theoretical Results

In this section, we prove convergence guarantees for `SPMA` in the multi-armed bandit and tabular MDP settings. In Section 4.1, we prove that `SPMA` with a constant step-size achieves linear convergence rates. In Section 4.2, we prove that SPMA with specific gap-dependent step-sizes achieves super-linear convergence for multi-armed bandits. We defer all the proofs to Appendices A and B.

4.1. Linear Convergence Rates

We first establish linear convergence for SPMA for multi-armed bandits for any constant $\eta \leq 1$.

Theorem 4.1 *The SPMA update in Eq. (3) with (i) a constant step-size $\eta \leq 1$ and (ii) uniform initialization i.e. $\pi_0(a) = \frac{1}{K}$ for all a results in the following convergence:*

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

where T is the number of iterations, a^* is the optimal arm and $\Delta_{\min} := \min_{a' \neq a^*} \Delta(a^*, a') = r(a^*) - r(a)$ is the gap.

The above theorem 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 the next theorem, we extend this 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. To state our theoretical result, we abuse the notation a little and use $J(\pi_t)$ to mean $J(z_t)$ where $\pi_t(a|s) \propto \exp(z_t(s,a))$.

Theorem 4.2 *Using the SPMA update in Eq. 2 with (i) a step-size $\eta < \min\left\{1 - \gamma, \frac{1}{C_t(1-\gamma)}\right\}$ results in the following convergence:*

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

where T is the number of iterations, π^* is the optimal policy, $C_t := \min_s\{\pi_t(\tilde{a}_t(s)|s)\, \Delta^t(s)\}$, $\tilde{a}_t(s) := \arg\max_a Q^{\pi_t}(s,a)$ and $\Delta^t(s) := \max_a Q^{\pi_t}(s,a) - \max_{a \neq \tilde{a}} Q^{\pi_t}(s,a)$.

For rewards in $(0,1)$, $C_t\,(1-\gamma) \in (0,1)$ and depends on the initialization. In order to put the above convergence result in context, note that SPG with a constant step-size results in a $\Theta\left({}^{1}/{}_{\epsilon}\right)$ convergence [15]. Recently, Chen et al. [6] prove 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. [13], Lu et al. [14] prove that SPG with adaptive step-sizes can 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 [12] making these rates vacuous. In contrast, the convergence result in Theorem 4.2 has no such dependence on the distribution mismatch ratio. The linear convergence rate in Theorem 4.2 matches that of NPG with a constant step-size [13] and compared to Liu et al. [13, Theorem 5.4], it results in a better dependence (exponential vs polynomial) on the gap $\Delta^t(s)$.

In the next section, we demonstrate that SPMA with specific gap-dependent step-sizes can achieve a global super-linear convergence rate for multi-armed bandits.

4.2. Super-linear Convergence Rates for Bandits

In order to achieve the desired fast rate of convergence, we modify the update in Eq. (3) 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:

$$\pi_{t+1}(a) = \pi_t(a)\ [1 + \sum_{a' \neq a} \pi_t(a')\, \eta_{a,a'}\, \Delta(a,a')] \tag{4}$$

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

Theorem 4.3 *Using the* SPMA *update in Eq. (4) with (i) $\eta_{a,a'} = \frac{1}{|\Delta(a,a')|}$ and a (ii) uniform initialization similar to Theorem 4.1 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'.

To the best of our knowledge, these are the first global super-linear rates for policy gradient methods on multi-armed bandit problems. In the next section, we evaluate SPMA empirically.

5. Empirical Evaluation

We evaluate SPMA on two types of problems: (i) tabular MDPs with access to the exact policy gradients, and (ii) MDPs with continuous state-actions spaces with inexact policy gradients. For the baselines, we consider mirror descent policy optimization (MDPO) [25], a generalization of PMD to handle function approximation. Note that MDPO is exactly equal to PMD (and hence NPG [9]) in the tabular setting. We also compare to constant step-size SPG [23]. For each of the continuous control Mujoco [24] tasks, we use the actor-critic architecture from stable baselines [18] and parameterize the policy using a neural network. In order to handle function approximation with SPMA, we follow the idea in Vaswani et al. [26] and describe the resulting approach in Appendix D. We present the tabular MDP results in Appendix C, while focusing on the continuous control tasks in the main paper.

5.1. Experimental Results

We follow the experimental protocol of Tomar et al. [25], running our experiments using 5 seeds and reporting the average results along with their 95% confidence intervals. To select η, we perform a grid search over 2×10^6 iterations and choose the η that provides the best area under the curve. We then use this same η for 2×10^7 iterations. The results in Fig. 1 suggest that SPMA can significantly outperform MDPO in the HalfCheetah and Ant environments, while the performance remains comparable for Hopper and Walker. For more details about the experimental setup please see Appendix E. By default, stable baselines [18] does not include the SPG surrogate, and our preliminary results show its performance is much worse than MDPO and SPMA.

6. Discussion

We analyzed the theoretical and empirical performance of SPMA, a softmax policy mirror ascent algorithm. For tabular MDPs, we demonstrated that SPMA outperforms SPG both theoretically and

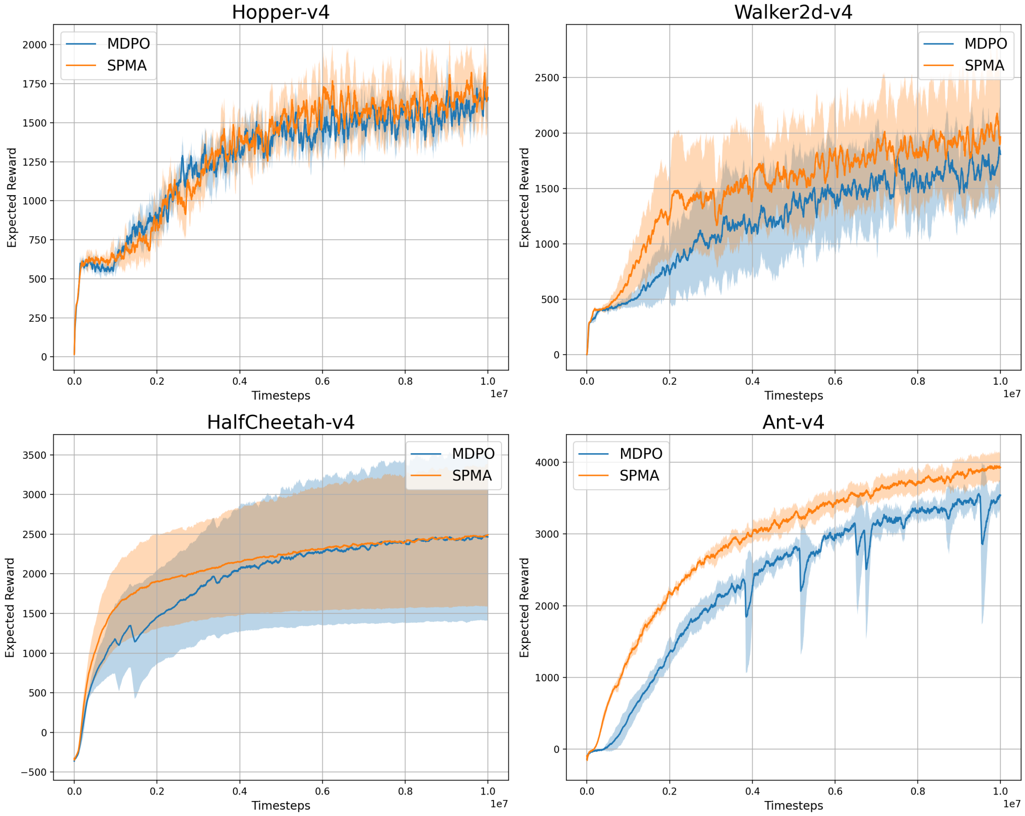

Figure 1: SPMA matches or outperforms MDPO across various MuJoCo environments.

empirically, while obtaining similar convergence guarantees and empirical performance as NPG. For continuous control tasks, we demonstrated that SPMA with function approximation can outperform MDPO, a generalization of NPG. In the future, we aim to prove convergence guarantees of SPMA with function approximation and inexact policy gradients and benchmark its performance against commonly used methods such as PPO [20] and TRPO [19].

References

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

[2] Larry Armijo. Minimization of functions having lipschitz continuous first partial derivatives. *Pacific Journal of mathematics*, 16(1):1–3, 1966.

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

[4] G Brockman. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[5] Nicolo Cesa-Bianchi, Yishay Mansour, and Gilles Stoltz. Improved second-order bounds for prediction with expert advice. *Machine Learning*, 66:321–352, 2007.

[6] Yen-Ju Chen, Nai-Chieh Huang, Ching-pei Lee, and Ping-Chun Hsieh. Accelerated policy gradient: On the convergence rates of the nesterov momentum for reinforcement learning. In *Forty-first International Conference on Machine Learning*, 2023.

[7] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

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

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

[10] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

[12] Gen Li, Yuting Wei, Yuejie Chi, Yuantao Gu, and Yuxin Chen. Softmax policy gradient methods can take exponential time to converge. In *Conference on Learning Theory*, pages 3107–3110. PMLR, 2021.

[13] Jiacai Liu, Wenye Li, and Ke Wei. Elementary analysis of policy gradient methods. *arXiv preprint arXiv:2404.03372*, 2024.

[14] Michael Lu, Matin Aghaei, Anant Raj, and Sharan Vaswani. Towards principled, practical policy gradient for bandits and tabular mdps. *arXiv preprint arXiv:2405.13136*, 2024.

[15] Jincheng Mei, Chenjun Xiao, Csaba Szepesvari, and Dale Schuurmans. On the global convergence rates of softmax policy gradient methods. In *International Conference on Machine Learning*, pages 6820–6829. PMLR, 2020.

[16] Jincheng Mei, Yue Gao, Bo Dai, Csaba Szepesvari, and Dale Schuurmans. Leveraging non-uniformity in first-order non-convex optimization. In *International Conference on Machine Learning*, pages 7555–7564. PMLR, 2021.

[17] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[18] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL http://jmlr.org/papers/v22/20-1364.html.

[19] John Schulman. Trust region policy optimization. *arXiv preprint arXiv:1502.05477*, 2015.

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

[21] Lior Shani, Yonathan Efroni, and Shie Mannor. Adaptive trust region policy optimization: Global convergence and faster rates for regularized mdps. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5668–5675, 2020.

[22] Richard S Sutton. Reinforcement learning: An introduction. *A Bradford Book*, 2018.

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

[24] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109.

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

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

[27] Sharan Vaswani, Amirreza Kazemi, Reza Babanezhad Harikandeh, and Nicolas Le Roux. Decision-aware actor-critic with function approximation and theoretical guarantees. *Advances in Neural Information Processing Systems*, 36, 2024.

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

Supplementary Material

Organization of the Appendix

Appendix A. Multi-armed Bandit Proofs

Theorem 4.1 *The* `SPMA` *update in Eq. (3) with (i) a constant step-size $\eta \leq 1$ and (ii) uniform initialization i.e. $\pi_0(a) = \frac{1}{K}$ for all a results in the following convergence:*

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

where T is the number of iterations, a^* is the optimal arm and $\Delta_{\min} := \min_{a' \neq a^*} \Delta(a^*, a') = r(a^*) - r(a)$ is the gap.

Proof . As in equation (3), we can write the update for arm a as following where $\Delta(a, a') = r(a) - r(a')$,

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

$$1 - \pi_{t+1}(a^*) = 1 - \pi_t(a^*) - \eta\, \pi_t(a^*) \left[\sum_{a' \neq a^*} \pi_t(a') \Delta(a^*, a')\right] \tag{5}$$

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

$$\begin{aligned} \sum_{a' \neq a^*} \pi_t(a') \Delta(a^*, a') &\geq \Delta_{\min} \sum_{a' \neq a^*} \pi_t(a') \\ &= \Delta_{\min}(1 - \pi_t(a^*)) \end{aligned} \tag{6}$$

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:

$$\pi_{t+1}(a^*) > \pi_t(a^*) > \ldots > \pi_0(a^*) = \frac{1}{K} \tag{7}$$

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

$$\begin{aligned}
\epsilon_{t+1} &= \epsilon_t - \eta\, \pi_t(a^*) \left[\sum_{a' \neq a^*} \pi_t(a') \Delta(a^*, a') \right] \\
&\leq \epsilon_t - \frac{\eta}{K} \left[\sum_{a' \neq a^*} \pi_t(a') \Delta(a^*, a') \right] && \text{(using (7))} \\
&\leq \epsilon_t - \frac{\eta \Delta_{\min}}{K} \epsilon_t && \text{(using (6))} \\
&= \epsilon_t (1 - \frac{\eta \Delta_{\min}}{K})
\end{aligned}$$

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

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

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

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

■

Theorem 4.3 *Using the* SPMA *update in Eq. (4) with (i) $\eta_{a,a'} = \frac{1}{|\Delta(a,a')|}$ and a (ii) uniform initialization similar to Theorem 4.1 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 $\pi_{t+1}(a)$:

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

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

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

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):

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

We use induction to show $\pi_t(a^*) = 1-\left[(1-\frac{1}{K})\right]^{2^t}$ 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):

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

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

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

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

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

■

Appendix B. MDP Proofs

Theorem 4.2 *Using the* `SPMA` *update in Eq. 2 with (i) a step-size* $\eta < \min\left\{1-\gamma, \frac{1}{C_t(1-\gamma)}\right\}$ *results in the following convergence:*

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

where T *is the number of iterations,* π^* *is the optimal policy,* $C_t := \min_s\{\pi_t(\tilde{a}_t(s)|s)\,\Delta^t(s)\}$, $\tilde{a}_t(s) := \arg\max_a Q^{\pi_t}(s,a)$ *and* $\Delta^t(s) := \max_a Q^{\pi_t}(s,a) - \max_{a\neq\tilde{a}} Q^{\pi_t}(s,a)$.

Proof First, we use the value difference Lemma to show the sMDPO update leads to a monotonic improvement in the value function.

$$J(\pi_{t+1}) - J(\pi_t) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\pi_{t+1}}}\left[\sum_a \pi_{t+1}(a|s)A^{\pi_t}(s,a)\right] \tag{8}$$

Plugging in the sMDPO update into the term in the bracket we obtain:

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

Hence $J(\pi_{t+1}) \geq J(\pi_t) \implies J_s(\pi_{t+1}) \geq J_s(\pi_t)$. We now compare sMDPO with a greedy update to show:

$$\eta\sum_a \pi_t(a|s)[A^{\pi_t}(s,a)]^2 \geq \eta\ C_t \max_a A^{\pi_t}(s,a) \tag{9}$$

Recall $\tilde{a}_t(s) := \arg\max_a A^{\pi_t}(s,a)$. We can split the sum on the LHS of the above over $\tilde{a}_t(s)$:

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

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

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

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

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

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

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

$$\begin{aligned}
\eta \sum_{a\neq\tilde{a}_t(s)} \pi_t(a|s)[A^{\pi_t}(s,a)]^2 &= \eta\ (1-\pi_t(\tilde{a}_t(s)|s))\ \mathbb{E}_{a\sim\tilde{\pi}_t}[A^{\pi_t}(s,a)]^2 \\
&\geq \eta\ (1-\pi_t(\tilde{a}_t(s)|s))\ \left(\mathbb{E}_{a\sim\tilde{\pi}_t}[A^{\pi_t}(s,a)]\right)^2 \ \ (\text{using } \mathbb{E}[x^2]\geq(\mathbb{E}[x])^2) \\
&= \eta\ (1-\pi_t(\tilde{a}_t(s)|s))\ \left(\mathbb{E}_{a\sim\tilde{\pi}_t}[A^{\pi_t}(s,a)]\right)\left(\mathbb{E}_{a\sim\tilde{\pi}_t}[A^{\pi_t}(s,a)]\right) \\
&= -\eta\ \pi_t(\tilde{a}_t(s)|s)A^{\pi_t}(s,\tilde{a}_t(s))\left(\mathbb{E}_{a\sim\tilde{\pi}_t}[A^{\pi_t}(s,a)]\right) \qquad (\text{using Eq. } 11) \\
&= -\eta\ \pi_t(\tilde{a}_t(s)|s)[\max_a A^{\pi_t}(s,a)]\left(\mathbb{E}_{a\sim\tilde{\pi}_t}[A^{\pi_t}(s,a)]\right) \\
&\qquad\qquad (\text{using the definition of } \tilde{a}_t(s))
\end{aligned}$$

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

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

So far we have shown:

$$\begin{aligned}
\sum_a \pi_{t+1}(a|s)A^{\pi_t}(s,a) &= \eta\sum_a \pi_t(a|s)[A^{\pi_t}(s,a)]^2 \\
&\geq \eta\ C_t \max_a A^{\pi_t}(s,a)
\end{aligned} \tag{12}$$

We now show a linear convergence when using the sMDPO update. 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 T at iteration t we observe:

$$TJ_s(\pi_t) - J_s(\pi_t) = \max_a Q^{\pi_t}(s,a) - J_s(\pi_t) = \max_a A^{\pi_t}(s,a) \tag{13}$$

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 $J_s(\pi')$ results in:

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

Using the above we obtain:

$$\begin{aligned}
T^{\pi_{t+1}} J_s(\pi_t) - J_s(\pi_t) &= \sum_a \pi_{t+1}(a|s) A^{\pi_t}(s,a) \\
&\geq \eta\ C_t \max_a A^{\pi_t}(s,a) && \text{(using Ineq. 12)} \\
&= \eta\ C_t \left[T J_s(\pi_t) - J_s(\pi_t)\right] && \text{(using Eq. 13)}
\end{aligned}$$

Assuming π^* is the optimal policy we have:

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

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

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

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

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

■

Appendix C. Tabular MDP Experiments

For the tabular MDP experiments, we use Cliff World [22] and Frozen Lake [4] as configured in [27] and initialize $z \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ uniformly , i.e., $\pi_0(a|s) = \frac{1}{|\mathcal{A}|}$ for each a and s. Furthermore, for each algorithm, we set η using a grid search and pick the ones that result in the best area under the curve. The tabular MDP results suggest SPMA and MDPO [25] achieve similar performance and they both outperform SPG [20, 23] (see Fig. 2). 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. 3 suggest that overall SPG [20, 23] (in green) is more sensitive to different values of η compared to SPMA (blue) and MDPO (red) [25]

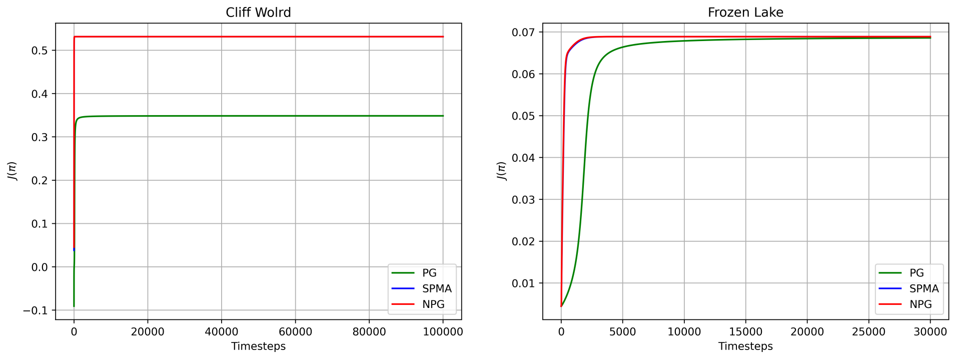

Figure 2: SPMA matches the performance of MDPO and they both outperform SPG.

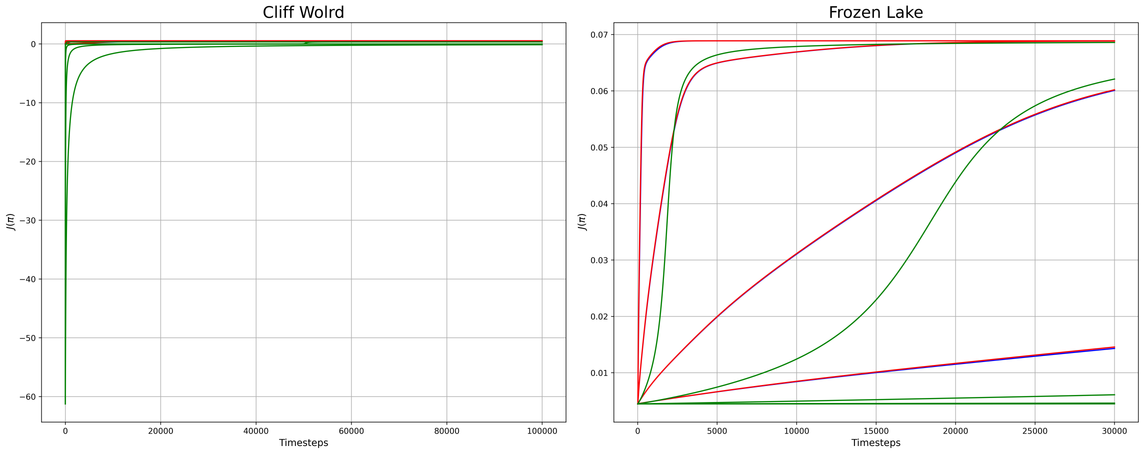

Figure 3: SPG's (green) is more sensitive to η compared to SPMA and MDPO (blue and red).

Appendix D. Handling Function Approximation

In order to incorporate function approximation, Vaswani et al. [26] distinguish between a policy's functional representation and parameterization. The *functional representation* of a policy defines its sufficient statistics. For example, for the *direct representation*, we represent a policy via the set of distributions $\pi(\cdot|s) \in \Delta_A$ for state $s \in \mathcal{S}$. On the other hand, for the *softmax representation*, we represent a policy by the set of logits $z^\pi(s,a)$ for each (s,a) pair such that $\pi(a|s) = {}^{\exp(z^\pi(s,a))}/{}_{\sum_{a'} \exp(z^\pi(s,a'))}$. Furthermore, the *policy parameterization* is determined by a *model* (with parameters θ) that realizes these statistics. For example, we could use a neural-network to parameterize the logits corresponding to the policy's softmax representation, rewriting $z^\pi(s,a) = z^\pi(s,a|\theta)$ where the model is implicit in the $z^\pi(s,a|\theta)$ notation. Note that the policy parameterization can be chosen independently of its functional representation and defines the set Π of realizable policies that can be expressed with the parametric model at hand.

In the special case of the tabular parameterization (no function approximation), a policy's representation is the same as it parameterization, which corresponds to having a parameter for each state-action pair and is equivalent to the setup analyzed in Section 4. Vaswani et al. [26] show that using functional mirror ascent with direct representation and negative entropy mirror map recovers the surrogate function $\ell_t(\theta)$ for `MDPO` [25] at iteration t:

$$\ell_t(\theta) = \mathbb{E}_{s\sim d^{\pi_t}} \left[\mathbb{E}_{a\sim\pi_t(.|s)} \left[\frac{\pi(a|s,\theta)}{\pi_t(a|s)} \left(\hat{Q}^{\pi_t}(s,a) - \frac{1}{\eta} \log \left(\frac{\pi(a|s,\theta)}{\pi_t(a|s)} \right) \right) \right] \right] ,$$

where $\hat{Q}^{\pi_t}(s,a)$ is the inexact Q function estimated via interacting with the environment. In the special case of tabular parameterization, the `MDPO` surrogate can be solved in closed form at each iteration and results in the `PMD` update. Similarly, using functional mirror ascent with the softmax representation and the log-sum-exp mirror map recovers the following surrogate function (referred to as `sMDPO` in Vaswani et al. [26]) at iteration t:

$$\ell_t(\theta) = \mathbb{E}_{s\sim d^{\pi_t}} \left[\mathbb{E}_{a\sim\pi_t(.|s)} \left[\left(\hat{A}^{\pi_t}(s,a) + \frac{1}{\eta} \right) \log \left(\frac{\pi(a|s,\theta)}{\pi_t(a|s)} \right) \right] \right] ,$$

where $\hat{A}^{\pi_t}(s,a)$ is the inexact advantage function estimated via interacting with the environment. In the special case of tabular parameterization, the above surrogate can be solved in closed form at each iteration and results in the `SPMA` update in Eq. (2).

In each iteration t, the agent first collects data, including visited states, actions taken, and rewards received, through interactions with the environment. This data is used to (i) construct a surrogate objective $\ell_t(\theta)$ for the actor-network and (ii) compute the mean squared error loss between the expected returns and the critic network's predictions. Both networks are optimized for m epochs in each iteration t.

Appendix E. Additional Details for Stable Baselines Experiments

We use $m = 5$ epochs in each iteration t of the optimization. For all hyper-parameters not mentioned in the main paper, we use their default values from stable baselines [18], except for the batch size and the Adam optimizer [10], which is used to optimize the surrogate objective for m epochs. Instead, we employ a full batch and minimize the surrogate in the deterministic setting using the Armijo line search [2]. This approach is motivated by two factors: (i) performing a grid search for the functional step size η is already computationally expensive, making an additional grid search for the inner loop step sizes infeasible, and (ii) this method yields better results compared to the default settings in stable baselines [18].