PERFORMATIVE POLICY GRADIENT: ASCENT TO OPTIMALITY IN PERFORMATIVE REINFORCEMENT LEARNING

Debabrota Basu, Udvas Das & Brahim Driss
Univ. Lille, Inria, CNRS,
Centrale Lille, UMR 9189 – CRIStAL, F-59000 Lille, France

Uddalak Mukherjee
ACMU, Indian Statistical Institute, Kolkata,
Dunlop, Baranagar, West Bengal 700108, India

ABSTRACT

Post-deployment machine learning algorithms often influence the environments they act in, and thus *shift* the underlying dynamics that the standard reinforcement learning (RL) methods ignore. While designing optimal algorithms in this *performative* setting has recently been studied in supervised learning, the RL counterpart remains under-explored. In this paper, we prove the performative counterparts of the performance difference lemma and the policy gradient theorem in RL, and further introduce the **Performative Policy Gradient** algorithm (PePG). PePG is the first policy gradient algorithm designed to account for performativity in RL. Under softmax parametrisation, and also with and without entropy regularisation, we prove that PePG converges to *performatively optimal policies*, i.e. policies that remain optimal under the distribution shifts induced by themselves. Thus, PePG significantly extends the prior works in Performative RL that achieves *performative stability* but not optimality. Furthermore, our empirical analysis on standard performative RL environments validate that PePG outperforms standard policy gradient algorithms and the existing performative RL algorithms aiming for stability.

1 INTRODUCTION

Reinforcement Learning (RL) studies the dynamic decision making problems under incomplete information (Sutton & Barto, 1998). Since an RL algorithm tries and optimises an utility function over a sequence of interactions with an unknown environment, RL has emerged as a powerful tool for algorithmic decision making. Specially, in the last decade, RL has underpinned some of the celebrated successes of AI, such as championing Go with AlphaGo (Silver et al., 2014), controlling particle accelerators (St. John et al., 2021), aligning Large Language Models (LLMs) (Bai et al., 2022), reasoning (Havrilla et al.), to name a few. But the existing paradigm of RL assumes that the underlying environment with which the algorithm interacts stays static over time and the goal of the algorithm is to find the utility-maximising, aka optimal policy for choosing actions over time for this specific environment. But *this assumption does not hold universally*.

In this digital age, algorithms are not passive. Their decisions also shape the environment they interact with, inducing distribution shifts. This phenomenon that predictive AI models often trigger actions that influences their own outcomes is termed as *performativity*. In the supervised learning setting, the study of *performative prediction* is pioneered by Perdomo et al. (2020), and then followed by an extensive literature encompassing stochastic optimisation, control, multi-agent RL, games (Izzo et al., 2021; 2022; Miller et al., 2021; Li & Wai, 2022; Narang et al., 2023; Piliouras & Yu, 2023; Góis et al., 2024; Barakat et al., 2025) etc. There has been several attempts to achieve performative optimality or stability for real-life tasks— recommendation systems (Eilat & Rosenfeld, 2023), measuring the power of firms (Hardt et al., 2022; Mofakhami et al., 2023), healthcare (Zhang et al., 2022) etc. Performativity of algorithms is also omnipresent in practically deployed RL systems. For example, an RL algorithm deployed in a recommender system does not only aim to maximise the user satisfaction but also shifts the preferences of the users in the long-term (Chaney et al., 2018; Mansoury et al., 2020). To clarify the impact of performativity, let us consider an example.

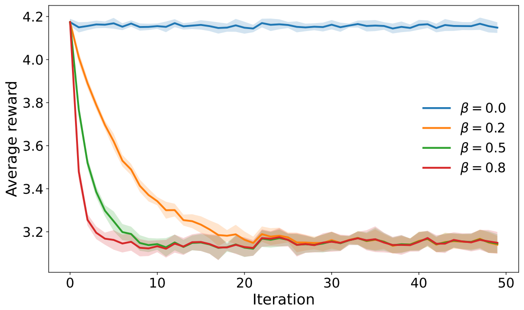

Figure 1: Average reward (over 10 runs) obtained by ERM and Performative Optimal policies across performative strength β.

Example 1 (Performative RL in loan approval)**.** *Let us consider a loan approval problem, where an applicant obtains a loan (or get rejected) according to their credit score x, and x depends on the capital of the applicant and that of the population. At each time t, a loan applicant arrives with a credit score x_t sampled from $\mathcal{N}(\mu_t, \sigma^2)$. The bank chooses whom to give a loan by applying a softmax binary classifier $\boldsymbol{\pi}_\theta : \mathbb{R} \rightarrow$*

$\{0,1\}$ *on x with threshold parameter θ. This decision has two effects. (a) The bank receives a positive payoff R, if the loan applicant who was granted a loan repays, or else, loses by L. Thus, the bank's expected utility for policy $\boldsymbol{\pi}_\theta$ is $U(\theta,\mu) = \mathbb{E}_{x\sim\mathcal{N}(\mu,\sigma^2)}\left[\boldsymbol{\pi}_{\boldsymbol{\theta}}(x)(\mathbb{P}(repayment|x)R - (1-\mathbb{P}(repayment|x))L)\right]$. (b) Since the amount of capital both the applicant and the population influence the credit score, we model that the change in the population mean μ_{t+1} depends on the bank's policy, via a grant rate $\mathbb{E}_{x\sim\mathcal{N}(\mu_t,\sigma_t^2)}\left[\pi_\theta(x)\right]$. Specifically, $\mu_{t+1} = (1-\beta)\mu_t + \beta f\left(\mathbb{E}_{x\sim\mathcal{N}(\mu_t,\sigma_t^2)}\left[\pi_\theta(x)\right]\right)$, where $\beta\in[0,1]$ is the performative strength and $f:\mathbb{R}\to[-M,M]$. Now, if one ignores the performative nature of this decision making problem, and try to find out the optimal with respect to a static credit distribution, it obtains $\theta^{ERM} \triangleq \arg\max_\theta U\big(\theta,\mu_0\big)$. In contrast, if it considers performativity, it obtains $\theta^{Perf} \triangleq \arg\max_\theta U\big(\theta,\mu^*(\theta)\big)$. In Figure 1, we show that the average reward obtained by both the solutions are significantly different. This demonstrates why performativity is a common phenomenon across algorithmic decision making problems, and how it changes the resulting optimal solution. Further details are in Appendix* **??**.

These problem scenarios have motivated the study of performative RL. Though Bell et al. (2021) were the first to propose a setting where the transition and reward of an underlying MDP depend non-deterministically on the deployed policy, Mandal et al. (2023) formally introduced *Performative RL*, and its solution concepts, i.e., performatively stable and optimal policies. Performative stable policies do not get affected or changed due to distribution shifts after deployment. Performatively optimal policies yield the highest expected return once deployed in the performative RL environment. Mandal et al. (2023) proposed direct optimization and ascent based techniques that attains performative stability upon repeated retraining. Extending this work, Rank et al. (2024) and Mandal & Radanovic (2024) manage to solve the same problem with delayed retraining for gradually shifting and linear MDPs. However, *there exists no algorithm yet in performative RL that provably converges to the performative optimal policy*.

As we know from the RL literature, the Policy Gradient (PG) type of algorithms that treats policy as a parametric function and updates the parameters through gradient ascent algorithms are efficient and scalable (Williams, 1992; Sutton et al., 1999; Kakade, 2001). Some examples of successful and popular policy gradient methods include TRPO (Schulman et al., 2015), PPO (Schulman et al., 2017), DDPG (Silver et al., 2014), SAC (Mnih et al., 2016), which are widely used in modern RL applications. Recent theoretical advances also establish finite-sample convergence guarantees and complexity analyses (Agarwal et al., 2021; Yuan et al., 2022) of PG algorithms. Motivated by the simplicity and universality of the PG algorithms, we ask these two questions in the context of performative RL:

1. *How to design PG-type algorithms for performative RL environments to achieve optimality?*
2. *What are the minimal conditions under which PG-type algorithms converge to the performatively optimal policy*?

Our contributions address these questions affirmatively, and showcases the difference of optimality-seeking and stability-seeking algorithms in performative RL.

I. Algorithm Design: We propose the first Performative Policy Gradient algorithm, PePG, for performative RL environments. Specifically, we extend the classical vanilla PG and entropy-regularised PG algorithms to Performative RL settings. Though the general algorithm design stays same, we derive a performative policy gradient theorem that shows, evaluation of the gradient involves two novel gradient terms in performative RL – (a) the expected gradient of reward, and (b) the expected gradient of log-transition probabilities times its impact on the expected cumulative return. We leverage this theorem to propose an estimator of the performative policy gradient under any differentiable parametrisation.

II. Convergence to Performative Optimality. We further analyse PePG (with and without entropy regularisation) for softmax policies, and softmax Performative Markov Decision Processes (PeMDPs), i.e. the MDPs with softmax transition probabilities and linear rewards with respect to the parameters of the softmax policy. We provide a minimal recipe to prove convergence of PePG using (a) smoothness of the performative value function, and (b) approximate gradient domination lemma for performative policy gradients. This allows us to show that PePG converges to an ϵ-ball around performative optimal policy in $\Omega\left(\frac{|\mathcal{S}||\mathcal{A}|^2}{\epsilon^2(1-\gamma)}\right)$ iterations, where $|\mathcal{S}|$ and $|\mathcal{A}|$ are the number of states and actions, respectively.

Specifically, Mandal et al. (2023) frames the question of using policy gradient to find stable policies as an open problem. The authors further contemplate, as PG functions in the policy space, whether it is possible to converge towards a stable policy. In this paper, we affirmatively solve an extension to this open problem for tabular softmax PeMDPs with softmax policies.

III. Stability- vs. Optimality-seeking Algorithms in Performative RL. We further theoretically and numerically contrast the performances of stability-seeking and optimality-seeking algorithms. Theoretically, we derive the performative performance difference lemma that distinguished the effect of policy update in these two types of algorithms. Numerically, we compare the performances of PePG with the state-of-the-art MDRR (Mixed Delayed Repeated Retraining (Rank et al., 2024)) algorithm for finding performatively stable policies in the multi-agent environment proposed by (Mandal et al., 2023). We show that PePG yields significantly higher values functions than MDRR, while MDRR achieves either similar or lower distance from stable state-action distribution than PePG .

2 Preliminaries: From RL to Performative RL

Now, we formalise the RL and performative RL problems, and provide the basics of policy gradient algorithms in RL.

2.1 RL: Infinite-horizon Discounted MDPs

In RL, we mostly study Markov Decision Processes (MDPs) defined via the tuple $(\mathcal{S}, \mathcal{A}, \mathbf{P}, r, \gamma)$, where $\mathcal{S} \subseteq \mathbb{R}^d$ is the state space and $\mathcal{A} \subseteq \mathbb{R}^d$ is the action space. Both the spaces are assumed to be compact. At any time step $t \in \mathbb{N}$, an agent plays an action $a_t \in \mathcal{A}$ at a state $s_t \in \mathcal{S}$. It transits the MDP environment to a state s_{t+1} according to a transition kernel $\mathbf{P}(\cdot \mid s_t, a_t) \in \Delta(\mathcal{S})$. The agent further receives a reward $r(s_t, a_t) \in \mathbb{R}$ quantifying the goodness of taking action a_t at s_t. The strategy to take an action is represented by a stochastic map, called *policy*, i.e. $\boldsymbol{\pi} : \mathcal{S} \to \Delta(\mathcal{A})$. Given an initial state distribution $\boldsymbol{\rho} \in \Delta(\mathcal{S})$, *the goal is to find the optimal policy* $\boldsymbol{\pi}^\star$ *that maximises* the expected discounted sum of rewards, i.e., the *value function*: $V_{\boldsymbol{\pi}}(\boldsymbol{\rho}) \triangleq \mathbb{E}_{s_0 \sim \boldsymbol{\rho}, s_{t+1} \sim \mathbb{P}(\cdot | s_t, \boldsymbol{\pi}(s_t))} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, \boldsymbol{\pi}(s_t))\right]$, where $\gamma \in (0, 1)$ is called the *discount factor*. γ indicates how much a previous reward matters in the next step, and bounds the effective horizon of a policy to $\frac{1}{1-\gamma}$.

Algorithm 1 Vanilla Policy Gradient

```
Input: Learning rate $\eta > 0$.
Initialize: Policy parameter $\boldsymbol{\theta}_0(s, a) \forall s \in \mathcal{S}, a \in \mathcal{A}$.
for $t = 1$ to T do
    Estimate the gradient $\nabla_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}}(\boldsymbol{\rho}) \mid_{\boldsymbol{\theta}=\boldsymbol{\theta}_t}$
    Gradient ascent step: $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t + \eta \nabla_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}}(\boldsymbol{\rho}) \mid_{\boldsymbol{\theta}=\boldsymbol{\theta}_t}$
end for
```

Policy Gradient (PG) Algorithms. PG-type algorithms maximise the value function by directly optimising the policy through a gradient over value function (Williams, 1992). To compute the gradient, we choose a parametric family of policies $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ for some $\boldsymbol{\theta} \in \mathbb{R}^d$ (e.g. direct (Agarwal et al., 2021; Wang & Zou, 2022), softmax (Agarwal et al., 2021; Mei et al., 2020), Gaussian (Ciosek & Whiteson, 2020; Ghavamzadeh & Engel, 2006)). Specifically, vanilla PG (Algorithm 1), performs a gradient ascent on the policy parameter at each step $t \in \mathbb{N}$. As the goal is to maximise $V^{\boldsymbol{\pi}}(\rho)$, we update $\boldsymbol{\theta}$ towards $\nabla_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}}(\rho)$, which is the direction improving the value $V^{\boldsymbol{\pi}}(\rho)$ with a fixed learning rate $\eta > 0$. For vanilla PG, the policy gradient takes the convenient form leading to estimators computable only with policy rollouts.

Theorem 1 (Policy Gradient Theorem (Sutton et al., 1999))**.** *Fix a differentiable paramterisation* $\theta \mapsto \pi_\theta(a \mid s)$ *and an initial distribution* $\boldsymbol{\rho}$*. Let us define the Q-value function* $Q^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(s, a) \triangleq \mathbb{E}_{s_{t+1} \sim \mathbf{P}_{\boldsymbol{\pi}}(\cdot | s_t, \boldsymbol{\pi}(s_t))} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, \boldsymbol{\pi}(s_t)) \mid s_0 = s, a_0 = a\right]$, *and advantage function* $A^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(s, a) \triangleq Q^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(s, a) - V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(s)$*. Then,*

$$\nabla_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{\rho}) = \frac{1}{1-\gamma} \mathbb{E}_{\tau \sim \mathbb{P}^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}} \left[\sum_{t=0}^{\infty} \gamma^t Q^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(s, a) \nabla_{\boldsymbol{\theta}} \log \boldsymbol{\pi}_{\boldsymbol{\theta}}(a \mid s)\right] = \mathbb{E}_{\tau \sim \mathbb{P}^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}} \left[\sum_{t=0}^{\infty} \gamma^t A^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(s, a) \nabla_{\boldsymbol{\theta}} \log \boldsymbol{\pi}_{\boldsymbol{\theta}}(a \mid s)\right].$$

Since the value function is not concave in the policy parameters, achieving optimality with PG has been a challenge. But practical scalability and efficiency of these algorithms has motivated a long-line of work to understand the minimum conditions and parametric forms of policies leading to convergence to the optimal policy (Agarwal et al., 2021; Mei et al., 2020; Wang & Zou, 2022; Yuan et al., 2022). *Our work extends these algorithmic techniques and theoretical insights to performative RL.*

2.2 Performative RL: Infinite-horizon Discounted PeMDPs

Given a policy set $\boldsymbol{\pi} \in \Pi$, we denote the Performative Markov Decision Process (PeMDP) is defined as the set of MDPs $\{\mathcal{M}(\boldsymbol{\pi}) \mid \boldsymbol{\pi} \in \Pi\}$, where each MDP is a tuple $\mathcal{M}(\boldsymbol{\pi}) \triangleq (\mathcal{S}, \mathcal{A}, \mathbf{P}_{\boldsymbol{\pi}}, r_{\boldsymbol{\pi}}, \gamma)$. Note, that the transition kernel and rewards distribution are no more invariant with respect to the policy. They shift with the deployed policy $\boldsymbol{\pi} \in \Delta(\mathcal{A})$ (Mandal et al., 2023; Mandal & Radanovic, 2024). In this setting, the probability of generating a trajectory $\tau_{\boldsymbol{\pi}} \triangleq (s_t, a_t)_{t=0}^{\infty}$ under policy $\boldsymbol{\pi}$ with underlying MDP $\mathcal{M}(\boldsymbol{\pi}')$ is given by[1] $\mathbb{P}^{\boldsymbol{\pi}}_{\boldsymbol{\pi}'}(\tau \mid \boldsymbol{\rho}) \triangleq \boldsymbol{\rho}(s_0) \prod_{t=0}^{\infty} \boldsymbol{\pi}(a_t \mid s_t) \mathbf{P}_{\boldsymbol{\pi}'}(s_{t+1} \mid s_t, a_t)$, where $\boldsymbol{\rho} \in \Delta(\mathcal{S})$ is the initial state distribution. Furthermore, the state-action occupancy measure for deployed policy $\boldsymbol{\pi}$ and environment-inducing policy $\boldsymbol{\pi}'$ is defined as $\boldsymbol{d}^{\boldsymbol{\pi}}_{\boldsymbol{\pi}', \boldsymbol{\rho}} \triangleq \frac{1}{1-\gamma} \mathbb{E}_{\tau \sim \mathbb{P}^{\boldsymbol{\pi}}_{\boldsymbol{\pi}'}} \left[\sum_{t=0}^{\infty} \gamma^t \mathbb{1}(s_t = s, a_t = a) \mid s_0 \sim \boldsymbol{\rho}\right]$. Now, we are ready to define the performative expected return, referred as the performative value function that we aim to maximise while solving PeMDP.

Definition 1 (Performative Value Function)**.** *Given a policy* $\boldsymbol{\pi} \in \Pi$ *and an initial state distribution* $\boldsymbol{\rho} \in \Delta(S)$*, the performative value function* $V^{\boldsymbol{\pi}}_{\boldsymbol{\pi}}(\boldsymbol{\rho})$ *is*

$$V^{\boldsymbol{\pi}}_{\boldsymbol{\pi}}(\boldsymbol{\rho}) \triangleq \mathbb{E}_{\tau \sim \mathbb{P}^{\boldsymbol{\pi}}_{\boldsymbol{\pi}}} \left[\sum_{t=0}^{\infty} \gamma^t r_{\boldsymbol{\pi}}(s_t, \boldsymbol{\pi}(s_t)) \mid s_0 \sim \boldsymbol{\rho}\right]. \tag{1}$$

[1] Hereafter, for relevant quantities, $\boldsymbol{\pi}$ in superscript denotes the deployed policy, and $\boldsymbol{\pi}'$ in the subscript denoted the environment-inducing, i.e. the policy inducing the transition kernel and reward function that the algorithm interacts with.

Equation (1) gives the total expected return that captures the performativity aspect in PeMDPs as the underlying dynamics changes with a deployed policy $\boldsymbol{\pi}(\cdot \mid s)$. Note that, we can maximise performative value function in two ways: (i) considering $\boldsymbol{\pi}$ as both the environment-inducing policy and the policy the RL agent deploys, or (ii) deploying $\boldsymbol{\pi}$ to fix it as the environment-inducing policy and agent plays another policy $\boldsymbol{\pi}'$. At this vantage point, let us introduce the notion of optimality and stability of policies in PeMDPs (Mandal et al., 2023).

Definition 2 (Performative Optimality)**.** *A policy $\boldsymbol{\pi}^\star_o$ is performatively optimal if it maximizes the performative value function.*

$$\boldsymbol{\pi}^\star_o \in \underset{\boldsymbol{\pi}\in\Delta(\mathcal{A})}{\arg\max}\, V^{\boldsymbol{\pi}}_{\boldsymbol{\pi}}(\boldsymbol{\rho})\,. \tag{2}$$

Thus, if we play the policy $\boldsymbol{\pi}$ in the environment induced by policy $\boldsymbol{\pi}$ to maximise the expected return, we land on the performatively optimal policy.

Definition 3 (Performative Stability)**.** *A policy $\boldsymbol{\pi}^\star_s$ is performatively stable if there is no gain in performative value function due to deploying any other policy than $\boldsymbol{\pi}^\star_s$ in the environment induced by $\boldsymbol{\pi}^\star_s$.*

$$\boldsymbol{\pi}^\star_s \in \underset{\boldsymbol{\pi}\in\Delta(\mathcal{A})}{\arg\max}\, V^{\boldsymbol{\pi}}_{\boldsymbol{\pi}^\star_s}(\boldsymbol{\rho}). \tag{3}$$

As noted by Mandal et al. (2023), a performatively optimal policy may not be performatively stable, i.e., $\boldsymbol{\pi}^\star_o$ may not be optimal for a changed underlying environment $\mathcal{M}(\boldsymbol{\pi}^\star_o)$, when it is deployed. Also, in general, the performative value function of $\boldsymbol{\pi}^\star_o$ might be equal to or higher than that of $\boldsymbol{\pi}^\star_s$. In this paper, *we design PG algorithms computing the performative optimal policy for a given set of MDPs*, and reinstate their differences with performatively stable policies.

The existing literature on PeMDPs (Mandal et al., 2023; Mandal & Radanovic, 2024; Rank et al., 2024; Pollatos et al., 2025; Chen et al., 2024) focused primarily on finding a performatively stable policy, i.e. a $\boldsymbol{\pi}^\star_s$ according to Definition 3. In practice, while the notion of stable policies matters for very specific applications, a stable policy may not always suffice. But they might show large sub-optimality gaps, which are often not desired for real-life tasks. *We fill up this gap in literature and propose the first provably converging and computationally efficient PG algorithm for PeMDPs.* Later on, we also empirically show the deficiency of the existing stability finding algorithms if we aim for optimality (Section 5).

Entropy Regularised PeMDPs. Entropy regularisation has emerged as a simple but powerful technique in classical RL to design smooth and efficient algorithms with sufficient exploration. Thus, we study another variant of the performative value function that is regularised using discounted entropy (Mei et al., 2020; Neu et al., 2017; Liu et al., 2019; Zhao et al., 2019). In this setting, the original value function in Definition 1 is regularised using the discounted entropy $H_{\boldsymbol{\pi}}(\rho) \triangleq \mathbb{E}_{\tau\sim\mathbb{P}^{\boldsymbol{\pi}}_{\boldsymbol{\pi}}}\left[-\sum_{t=0}^{\infty}\gamma^t \log\boldsymbol{\pi}(a_t \mid s_t)\right]$. This is equivalent to maximising the expected reward with a shifted reward function $\tilde{r}_{\boldsymbol{\pi}}(\boldsymbol{\pi}(s_t), s_t) = r_{\boldsymbol{\pi}}(\boldsymbol{\pi}(s_t), s_t) - \lambda\log(\boldsymbol{\pi}(a_t \mid s_t))$ for some $\lambda \geq 0$. $\tilde{r}_{\boldsymbol{\pi}}$ is referred as the "soft-reward" in MDP literature (Wang & Uchibe, 2024; Herman et al., 2016; Shi et al., 2019). This allows us to define the soft performative value function.

Definition 4 (Entropy Regularised (or Soft) Performative Value Function)**.** *Given a policy $\boldsymbol{\pi} \in \Pi$, a starting state distribution $\boldsymbol{\rho} \in \Delta(S)$, and a regularisation parameter $\lambda \geq 0$, the soft performative value function $V^{\boldsymbol{\pi}}_{\boldsymbol{\pi}}(\boldsymbol{\rho})$ is*

$$\tilde{V}^{\boldsymbol{\pi}}_{\boldsymbol{\pi}}(\boldsymbol{\rho}) \triangleq \mathbb{E}_{\tau\sim\mathbb{P}^{\boldsymbol{\pi}}_{\boldsymbol{\pi}}}\left[\sum_{t=0}^{\infty}\gamma^t\left(r_{\boldsymbol{\pi}}(s_t,\boldsymbol{\pi}(s_t)) - \lambda\log\boldsymbol{\pi}(a_t\mid s_t)\right) \mid s_0\sim\boldsymbol{\rho}\right] = \mathbb{E}_{\tau\sim\mathbb{P}^{\boldsymbol{\pi}}_{\boldsymbol{\pi}}}\left[\sum_{t=0}^{\infty}\gamma^t\tilde{r}_{\boldsymbol{\pi}}(s_t,\boldsymbol{\pi}(s_t)) \mid s_0\sim\boldsymbol{\rho}\right]. \tag{4}$$

Since policies belong to the probability simplex, the entropy regularisation naturally lends to smoother and stable PG algorithms. Later, we show that the discounted entropy is a smooth function of the policy parameters for PeMDPs extending the optimization-wise benefits of entropy regularisation to PeMDPs. Additionally, using the notion of soft rewards, we can further define soft performatively optimal and stable policies for entropy regularised PeMDPs. Leveraging it, *we unifiedly design PG algorithms for both the unregularised and the entropy regularised PeMDPs.*

3 Policy Gradient Algorithms in Performative RL

In this section, we first study the impact of policy updates in PeMDPs. Then, we leverage it to derive the performative policy gradient theorem and design Performative Policy Gradient (PePG) algorithm for any differentiable parametric policy class.

3.1 Impact of Policy Updates on PeMDPs

Performance difference lemma has been central in RL to understand the impact of changing policies in terms of value functions (Kakade & Langford, 2002). It has been also central to analysing and developing PG-type methods (Agarwal et al., 2021; Silver et al., 2014; Kallel et al., 2024). But the existing versions of performance difference cannot handle performativity. Here, we derive the performative version of the performance difference lemma that quantifies the shift in the performative value function due to change the deployed and environment-inducing policies.

Lemma 1 (Performative Performance Difference Lemma)**.** *The difference in performative value functions induced by* $\boldsymbol{\pi}$ *and* $\boldsymbol{\pi}' \in \Pi$ *while starting from the initial state distribution* $\boldsymbol{\rho}$ *is*

$$V^{\boldsymbol{\pi}}_{\boldsymbol{\pi}}(\boldsymbol{\rho}) - V^{\boldsymbol{\pi}'}_{\boldsymbol{\pi}'}(\boldsymbol{\rho}) = \frac{1}{1-\gamma}\mathbb{E}_{(s,a)\sim \boldsymbol{d}^{\boldsymbol{\pi}}_{\boldsymbol{\pi}',\rho}}[A^{\boldsymbol{\pi}'}_{\boldsymbol{\pi}'}(s,a)] + \frac{1}{1-\gamma}\mathbb{E}_{(s,a)\sim \boldsymbol{d}^{\boldsymbol{\pi}}_{\boldsymbol{\pi},\rho}}\left[(r_{\boldsymbol{\pi}}(s,a) - r_{\boldsymbol{\pi}'}(s,a)) + \gamma(\mathbf{P}_{\boldsymbol{\pi}}(\cdot|s,a) - \mathbf{P}_{\boldsymbol{\pi}'}(\cdot|s,a))^\top V^{\boldsymbol{\pi}}_{\boldsymbol{\pi}}(\cdot)\right]. \tag{5}$$

where $A^{\boldsymbol{\pi}'}_{\boldsymbol{\pi}'}(s,a) \triangleq Q^{\boldsymbol{\pi}'}_{\boldsymbol{\pi}'}(s,a) - V^{\boldsymbol{\pi}'}_{\boldsymbol{\pi}'}(s)$ *is the performative advantage function for any state* $s \in \mathcal{S}$ *and action* $a \in \mathcal{A}$*.*

The crux of the proof is decomposing the performative value through environment-inducing and deployed policies

$$V^{\boldsymbol{\pi}}_{\boldsymbol{\pi}}(s_0) - V^{\boldsymbol{\pi}'}_{\boldsymbol{\pi}'}(s_0) = \underbrace{V^{\boldsymbol{\pi}}_{\boldsymbol{\pi}}(s_0) - V^{\boldsymbol{\pi}}_{\boldsymbol{\pi}'}(s_0)}_{\text{performative shift term}} + \underbrace{V^{\boldsymbol{\pi}}_{\boldsymbol{\pi}'}(s_0) - V^{\boldsymbol{\pi}'}_{\boldsymbol{\pi}'}(s_0)}_{\text{performance difference term}}.$$

(1) *Connection to Classical RL.* In classical RL, the performance difference lemma yields $V^{\boldsymbol{\pi}}(\boldsymbol{\rho}) - V^{\boldsymbol{\pi}'}(\boldsymbol{\rho}) = \frac{1}{1-\gamma}\mathbb{E}_{(s,a)\sim \boldsymbol{d}^{\boldsymbol{\pi}}_{\rho}}[A^{\boldsymbol{\pi}'}(s,a)]$. The first term in Lemma 1 is equivalent to the classical result in the environment induced by $\boldsymbol{\pi}'$. But due to environment shift, two more terms appear in the performative performance difference incorporating the impacts of reward shifts and transition shifts. (2) *Connection to Performative Stability.* If we ignore the reward and transition shift terms, the performance difference term $V^{\boldsymbol{\pi}}_{\boldsymbol{\pi}'}(s_0) - V^{\boldsymbol{\pi}'}_{\boldsymbol{\pi}'}(s_0)$ quantifies the impact of changing the deployed policy from $\boldsymbol{\pi}'$ to $\boldsymbol{\pi}$ in an environment induced by $\boldsymbol{\pi}'$. Thus, a stability seeking algorithm would like to minimise this term, while an optimality seeking algorithm has to incorporate all of the terms.

Now, we ask: *how much do the new environment shift terms change the performative performance difference?*

For simplicity, we focus on the commonly studied PeMDPs with bounded rewards and gradually shifting environments, i.e. the ones with Lipschitz transitions and rewards with respect to the deployed policies (Rank et al., 2024).

Assumption 1 (Bounded reward)**.** *We assume that the rewards are bounded in* $[-R_{\max}, R_{\max}]$*.*

This is the only assumption needed through the paper and is standard in MDP literature (Mei et al., 2020; Li & Yang, 2023).

Lemma 2 (Bounding Performative Performance Difference for Gradually Shifting Environments)**.** *Let us assume that both rewards and transitions are Lipschitz functions of policy, i.e.* $\|r_{\boldsymbol{\pi}} - r_{\boldsymbol{\pi}'}\| \leq L_r \|\boldsymbol{\pi} - \boldsymbol{\pi}'\|$ *and* $\|\mathbf{P}_{\boldsymbol{\pi}} - \mathbf{P}_{\boldsymbol{\pi}'}\| \leq L_{\mathbf{P}} \|\boldsymbol{\pi} - \boldsymbol{\pi}'\|$, *for some* $L_r, L_{\mathbf{P}} \geq 0$*. Then, under Assumption 1, the performative shift in the sub-optimality gap of a policy* $\boldsymbol{\pi_\theta}$ *satisfies*

$$\left|V^{\boldsymbol{\pi}^\star_o}_{\boldsymbol{\pi}^\star_o}(\boldsymbol{\rho}) - V^{\boldsymbol{\pi_\theta}}_{\boldsymbol{\pi_\theta}}(\boldsymbol{\rho}) - \frac{1}{1-\gamma}\mathbb{E}_{(s,a)\sim d^{\boldsymbol{\pi}^\star_o}_{\boldsymbol{\pi_\theta},\boldsymbol{\rho}}}[A^{\boldsymbol{\pi_\theta}}_{\boldsymbol{\pi_\theta}}(s,a)]\right| \leq \frac{2\sqrt{2}}{1-\gamma}(L_r + \frac{\gamma}{1-\gamma}L_{\mathbf{P}}R_{\max})\mathbb{E}_{s_0\sim\boldsymbol{\rho}}D_{\mathrm{H}}\left(\boldsymbol{\pi}^\star_o(\cdot|s_0)\|\boldsymbol{\pi_\theta}(\cdot|s_0)\right). \tag{6}$$

where $D_{\mathrm{H}}(\mathbf{x}\|\mathbf{y})$ *denotes the Hellinger distance between* $\mathbf{x}$ *and* $\mathbf{y}$*.*

Implication. Lemma 2 shows novel characterisation of the *extra cost* we have to pay to adapt to performativity of the environment in terms of Hellinger distance between the true performatively optimal policy $\boldsymbol{\pi}^\star_o$ and any other parametrised policy $\boldsymbol{\pi_\theta}$. This implies that the order of difference between the optimal performative value function and that of any stability-seeking algorithm is $\Theta(\frac{1}{1-\gamma})$. This significantly improves the known order of sub-optimality achieved by existing algorithms. Specifically, Mandal et al. (2023) show that using repeated policy optimisation algorithms converges to a suboptimality gap $\mathcal{O}\left(\max\{\frac{S^{5/3}A^{1/3}\epsilon^{2/3}}{(1-\gamma)^{14/3}}, \frac{\epsilon S}{(1-\gamma)^4}\}\right)$. Thus, we see an opportunity to improve on the existing works and design algorithms that can achieve suboptimality gap of order $\Theta(\frac{1}{1-\gamma})$.

3.2 Algorithm Design: Performative Policy Gradient (PePG)

To achieve performative optimality, the goal is to maximise value function at the end of learning process. Gradient ascent is a standard first-order optimisation method to find maxima of a function. Similar to Algorithm 1, the crux of performative policy gradient method lies in the ascent step:

$$\boldsymbol{\theta}_{t+1} \leftarrow \begin{cases} \boldsymbol{\theta}_t + \eta_t \nabla_{\boldsymbol{\theta}} V^{\boldsymbol{\pi_\theta}}_{\boldsymbol{\pi_\theta}}(\tau)\ |_{\boldsymbol{\theta}=\boldsymbol{\theta}_t} & \text{, for unregularised objective} \\ \boldsymbol{\theta}_t + \eta_t \nabla_{\boldsymbol{\theta}} \tilde{V}^{\boldsymbol{\pi_\theta}}_{\boldsymbol{\pi_\theta}}(\tau)\ |_{\boldsymbol{\theta}=\boldsymbol{\theta}_t} & \text{, for Entropy-regularised objective.} \end{cases} \tag{7}$$

Given this ascent step, we have to evaluate the gradient at each time step from the rollouts of the present policy. In classical PG, the policy gradient theorem serves this purpose (Williams, 1992; Sutton et al., 1999; Silver et al., 2014). Thus, we derive the performative counterpart of the classic policy gradient theorem.

Algorithm 2 PePG: **Pe**rformative **P**olicy **G**radient

```
1: Input: Transition Feature Map $\psi(s)\forall s \in \mathcal{S}, \xi \in [-R_{\max}, R_{\max}]$ and discount factor $\gamma$.
2: Initialize: Initial policy parameters $\theta_0$, initial value function parameters $\phi_0$
3: for $k = 1, 2, \ldots$ do
4:     Collect trajectories: $\mathcal{D}_k = \{\tau_i\}_{i=1}^{I}$, where each $\tau_i \triangleq \{(s_{i,t}, a_{i,t}, s_{i,t+1}, r_{i,t})\}_{t=0}^{T-1}$ by playing $\boldsymbol{\pi}_{\boldsymbol{\theta}_k} = \boldsymbol{\pi}(\boldsymbol{\theta}_k)$
5:     Compute returns $R_k \triangleq \{R_{k,i}\}_{i=1}^{I}$, where $R_{k,i} = \{R_{k,i,t}\}_{t=0}^{T-1}$
6:     Compute advantage estimates $\hat{A}_k(\tau_i)$ using value function $\hat{V}_{\phi_k}(\tau_i)$ for each $\tau_i \in \mathcal{D}_k$ (estimate of $V^{\boldsymbol{\pi}_{\boldsymbol{\theta}_k}}_{\boldsymbol{\pi}_{\boldsymbol{\theta}_k}}(\tau_i)$ obtained
       from fitted value network with parameters $\phi_k$)
7:     Gradient estimation: Estimate policy gradient using (10)
8:     Gradient ascent step: Update policy parameters using (7)
9:     Fit value function $V_{\phi_{k+1}}$:
           $$\phi_{k+1} \leftarrow \arg\min_{\phi} \frac{1}{I \cdot T} \sum_{i=1}^{I} \sum_{t=0}^{T-1} \left(\hat{V}_{\phi_k}(s_t \in \tau_i) - R_{k,i,t}\right)^2$$
10: end for
```

Theorem 2 (Performative Policy Gradient Theorem)**.** *The gradient of the performative value function w.r.t $\boldsymbol{\theta}$ is as follows:*

(a) For unregularised objective,

$$\nabla_{\boldsymbol{\theta}} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\tau) = \mathbb{E}_{\tau \sim \mathbb{P}^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}} \left[\sum_{t=0}^{\infty} \gamma^t \left(A^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(s_t, a_t) \left(\nabla_{\boldsymbol{\theta}} \log \boldsymbol{\pi}_{\boldsymbol{\theta}}(a_t \mid s_t) + \nabla_{\boldsymbol{\theta}} \log P_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(s_{t+1}|s_t, a_t) \right) + \nabla_{\boldsymbol{\theta}} r_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(s_t, a_t) \right) \right], \quad (8)$$

(b) For entropy-regularized objective, we define $\tilde{A}^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(s, a) = Q^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(s, a) - V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(s) - \lambda \log \boldsymbol{\pi}_{\boldsymbol{\theta}}(a|s)$*, and get*

$$\nabla_{\boldsymbol{\theta}} \tilde{V}^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\tau) = \mathbb{E}_{\tau \sim \mathbb{P}^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}} \left[\sum_{t=0}^{\infty} \gamma^t \left(\tilde{A}^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(s_t, a_t) \left(\nabla_{\boldsymbol{\theta}} \log \boldsymbol{\pi}_{\boldsymbol{\theta}}(a_t \mid s_t) + \nabla_{\boldsymbol{\theta}} \log P_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(s_{t+1}|s_t, a_t) \right) + \nabla_{\boldsymbol{\theta}} \tilde{r}_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(s_t, a_t|\boldsymbol{\theta}) \right) \right]. \quad (9)$$

PePG**:** To elaborate on the design of PePG (Algorithm 2), we focus only on the REINFORCE update and softmax policy parametrisation. With the appropriate parameter choices, and initialisation of the policy parameter $\boldsymbol{\theta}$ and value function parameter ϕ, for each episode $k = 1, 2, \ldots$, PePG collects I trajectories to calculate return R^i and estimates advantage function $\hat{A}_k$ (Line 4-6). For a particular trajectory τ_i, the estimated advantage for a given state-action is $\widehat{A^{\boldsymbol{\pi}_{\boldsymbol{\theta}_k}}_{\boldsymbol{\pi}_{\boldsymbol{\theta}_k}}(s^i_t, a^i_t)} = R^i_{t,k} - V_{\phi_k}(s^i_t)$, where $R^i = \sum_{t=0}^{T-1} \gamma^t r_{\boldsymbol{\pi}_{\boldsymbol{\theta}_k}}(s^i_t, a^i_t)$.

Gradient Estimation (Line 7). With the necessary estimates in hand for all the collected I trajectories, PePG computes average gradient estimate over all the trajectories using

$$\widehat{\nabla_{\boldsymbol{\theta}_k} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}_k}}_{\boldsymbol{\pi}_{\boldsymbol{\theta}_k}}} = \frac{1}{I} \sum_{i=1}^{I} \sum_{t=0}^{T} \gamma^t (\widehat{A^{\boldsymbol{\pi}_{\boldsymbol{\theta}_k}}_{\boldsymbol{\pi}_{\boldsymbol{\theta}_k}}(s^i_t}, a^i_t) \left(\nabla_{\boldsymbol{\theta}_k} \log \boldsymbol{\pi}_{\boldsymbol{\theta}_k}(a^i_t \mid s^i_t) + \nabla_{\boldsymbol{\theta}_k} \log P_{\boldsymbol{\pi}_{\boldsymbol{\theta}_k}}(s^i_{t+1}|s^i_t, a^i_t) \right) + \nabla_{\boldsymbol{\theta}_k} r_{\boldsymbol{\pi}_{\boldsymbol{\theta}_k}}(s^i_t, a^i_t|\boldsymbol{\theta}_k)) \quad (10)$$

where all the individual gradients $\nabla_{\boldsymbol{\theta}_k} \log P_{\boldsymbol{\pi}_{\boldsymbol{\theta}_k}}, \nabla_{\boldsymbol{\theta}_k} r_{\boldsymbol{\pi}_{\boldsymbol{\theta}_k}}$ and $\nabla_{\boldsymbol{\theta}_k} \log \boldsymbol{\pi}_{\boldsymbol{\theta}_k}$ have the closed form expressions for softmax parametrisation according to Equation (12). Further, in Line 8, PePG updates the policy parameter for the next episode using a gradient ascent step leveraging the estimated average gradient over all I trajectories. Specifically, we plug in $\widehat{\nabla_{\boldsymbol{\theta}_k} V^{\boldsymbol{\pi}_{\boldsymbol{\theta}_k}}_{\boldsymbol{\pi}_{\boldsymbol{\theta}_k}}}$ to both the unregularised and entropy-regularised update rules are given in Equation (7). For the next episode, we again run a regression to update the value network plugging in the current estimates and resume the learning process further.

4 Convergence Analysis of PePG: Softmax Policies and Softmax PeMDPs

For rigorous theoretical analysis of PePG, we restrict ourselves to *softmax policy class*, and *softmax PeMDPs*. We define the softmax PeMDPs as the ones having softmax transition kernels with feature map $\psi(\cdot) : \mathcal{S} \to \mathbb{R}$, and linear reward functions with respect to the policy parameters, for all state $s \in \mathcal{S}$ and action $a \in \mathcal{A}$. Specifically, the class of softmax PeMDPs is $\{\mathcal{M}(\boldsymbol{\theta}) = \mathcal{M}(\boldsymbol{\pi_\theta}) \mid \boldsymbol{\theta} \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}\}$ such that

$$\boldsymbol{\pi_\theta}(a|s) = \frac{e^{\boldsymbol{\theta}_{s,a}}}{\sum_{a'} e^{\boldsymbol{\theta}_{s,a'}}}\,, \mathbf{P}_{\boldsymbol{\pi_\theta}}(s'|s,a) = \frac{e^{\boldsymbol{\theta}_{s,a}\psi(s')}}{\sum_{s''} e^{\boldsymbol{\theta}_{s,a}\psi(s'')}}\,, r_{\boldsymbol{\pi_\theta}}(s,a) = \mathcal{P}_{[-R_{\max},R_{\max}]}[\xi\boldsymbol{\theta}_{s,a}]\,, \tag{11}$$

where ψ is non-negative and upper bounded by $\psi_{\max} > 0$, and $\xi \in [0, R_{\max}]$ to align with Assumption 1.

Thus, we derive the derivatives of policy, transitions, and rewards as

$$\begin{gathered}\frac{\partial}{\partial\boldsymbol{\theta}_{s',a'}}\log\boldsymbol{\pi_\theta}(a|s) = \mathbb{1}[s=s',a=a'] - \boldsymbol{\pi_\theta}(a'|s)\mathbb{1}[s=s'],\\ \frac{\partial}{\partial\boldsymbol{\theta}_{s',a'}}\log\mathbf{P}_{\boldsymbol{\pi_\theta}}(s''|s,a) = \psi(s'')\mathbb{1}[s=s',a=a']\left(1-\mathbf{P}_{\boldsymbol{\pi_\theta}}(s''|s,a)\right)\,, \frac{\partial}{\partial\boldsymbol{\theta}_{s',a'}} r_{\boldsymbol{\pi_\theta}}(s,a) = \xi\mathbb{1}[s=s',a=a']\,.\end{gathered} \tag{12}$$

Given the derivatives, we can now readily estimate the policy gradient and deploy PePG for softmax PeMDPs.

Convergence Analysis: Challenges and Three Step Analysis. The main challenge to prove convergence of PePG is that the performative value function is not concave in the paramterisation $\boldsymbol{\theta}$, in general, and also in softmax PeMDPs. The similar issue occurs while proving convergence of PG-type algorithms in classical RL, which has been overcome by leveraging smoothness properties of the value functions and by deriving the local Polyak-Lojasiewicz (PL)-type conditions, known as *gradient domination*, with respect to the policy paramterisation. Leveraging these insights, we devise a three step convergence analysis for PePG.

Step 1: Smoothness of Performative Value Functions. First, we prove that the unregularised performative value function is $\mathcal{O}(\frac{|\mathcal{A}|}{(1-\gamma)^2})$ smooth. As we show that the entropy is also a smooth function for softmax PeMDPs, then under proper choice of the regaularisation parameter, i.e., $\lambda = \frac{1-\gamma}{1+2\log|\mathcal{A}|}$, entropy regularised performative value function is also $\mathcal{O}(\frac{|\mathcal{A}|}{(1-\gamma)^2})$ smooth. Since gradient ascent/descent methods can work well in smooth functions, we proceed thoroughly.

Step 2: Gradient Domination for Softmax PeMDPs. Now, the next step is to relate the performative performance difference with the performative policy gradient. This allows us to connect the per iteration improvement in the performative value function with the performative gradient descent at that step. These are known as PL-type inequalities. For non-concave objectives, PL inequalities guarantee convergence to global maxima by showing that the gradient of the objective at any parameter dominates the sub-optimality w.r.t. that parameter.

Lemma 3 (Performative Gradient Domination for Softmax PeMDPs)**.** *For PeMDPs defined in (11) and $C_\psi > -1$, we get*

(a) For unregularised value function:

$$V_{\boldsymbol{\pi}_o^\star}^{\boldsymbol{\pi}_o^\star}(\rho) - V_{\boldsymbol{\pi_\theta}}^{\boldsymbol{\pi_\theta}}(\rho) \leq \frac{\sqrt{|\mathcal{S}||\mathcal{A}|}}{(1+C_\psi)}\left\|\frac{\boldsymbol{d}_{\boldsymbol{\pi_\theta},\rho}^{\boldsymbol{\pi}_o^\star}}{\boldsymbol{d}_{\boldsymbol{\pi_\theta},\nu}^{\boldsymbol{\pi_\theta}}}\right\|_\infty \|\nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\pi_\theta}}^{\boldsymbol{\pi_\theta}}(\nu)\|_2 + 2\sqrt{2}\frac{R_{\max}}{1-\gamma}(|\mathcal{A}| + \frac{\gamma}{1-\gamma}\psi_{\max})\,. \tag{13}$$

(b) For entropy-regularised value function, $\tilde{V}_{\boldsymbol{\pi}_o^\star}^{\boldsymbol{\pi}_o^\star}(\rho) - \tilde{V}_{\boldsymbol{\pi_\theta}}^{\boldsymbol{\pi_\theta}}(\rho) \leq$

$$\frac{\sqrt{|\mathcal{S}||\mathcal{A}|}}{(1+C_\psi)}\left\|\frac{\boldsymbol{d}_{\boldsymbol{\pi_\theta},\rho}^{\boldsymbol{\pi}_o^\star}}{\boldsymbol{d}_{\boldsymbol{\pi_\theta},\nu}^{\boldsymbol{\pi_\theta}}}\right\|_\infty \|\nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\pi_\theta}}^{\boldsymbol{\pi_\theta}}(\nu)\|_2 + \frac{2\sqrt{2}}{1-\gamma}\left(R_{\max}|\mathcal{A}| + \frac{\gamma}{1-\gamma}\psi_{\max}(R_{\max} + \lambda\log|\mathcal{A}|)\right) + \frac{\lambda}{1-\gamma}(1+2\log|\mathcal{A}|) \tag{14}$$

Step 3: Iterative Application of Gradient Domination for Smooth Functions. Now, we can apply gradient domination along with the classic iterative convergence proof of gradient ascent for smooth functions. The intuition is that since the per-step sub-optimality is dominated by the gradient and the smooth functions are bounded by quadratic envelopes of parameters, applying gradient ascent iteratively would bring the sub-optimality down to small error level after enough iterations. We formalise this in Theorem 3.

Theorem 3 (Convergence of PePG in softmax PeMDPs)**.** *Let* $\mathrm{Cov} \triangleq \max_{\boldsymbol{\theta},\nu}\left\|\frac{\boldsymbol{d}_{\boldsymbol{\pi_\theta},\rho}^{\boldsymbol{\pi}_o^\star}}{\boldsymbol{d}_{\boldsymbol{\pi_\theta},\nu}^{\boldsymbol{\pi_\theta}}}\right\|_\infty$. *The gradient ascent algorithm on $V_{\boldsymbol{\pi_\theta}}^{\boldsymbol{\pi_\theta}}(\rho)$ (Equation (7)) with step size $\eta = \Omega((1-\gamma)^2/|\mathcal{A}|)$ satisfies, for all distributions $\rho \in \Delta(\mathcal{S})$,*

(a) For unregularised case, $\min_{t<T}\left\{V_{\boldsymbol{\pi}_o^\star}^{\boldsymbol{\pi}_o^\star}(\rho) - V_{\boldsymbol{\pi}_{\boldsymbol{\theta}_t}}^{\boldsymbol{\pi}_{\boldsymbol{\theta}_t}}(\rho)\right\} \leq \epsilon$ *when* $T = \Omega\left(\frac{R_{\max}|\mathcal{S}||\mathcal{A}|^2}{\epsilon^2(1-\gamma)^3}\mathrm{Cov}^2\right)$, *and* $\epsilon = \Omega\left(\frac{1}{1-\gamma}\right)$.

(b) For entropy regularised case, if we set $\lambda = \frac{(1-\gamma)}{1+2\log|\mathcal{A}|}$, *we get* $\min_{t<T}\left\{\tilde{V}_{\boldsymbol{\pi}_o^\star}^{\boldsymbol{\pi}_o^\star}(\rho) - \tilde{V}_{\boldsymbol{\pi}_{\boldsymbol{\theta}}^{(t)}}^{\boldsymbol{\pi}_{\boldsymbol{\theta}}^{(t)}}(\rho)\right\} \leq \epsilon$ *when* $T = \Omega\left(\frac{R_{\max}|\mathcal{S}||\mathcal{A}|^2}{\epsilon^2(1-\gamma)^3}\mathrm{Cov}^2\right)$, *and* $\epsilon = \Omega\left(\frac{1}{1-\gamma}\right)$.

Algorithms	Regulariser λ	Min. #samples	Environment
RPO FS (Mandal et al., 2023)	$\mathcal{O}\left(\frac{\lvert\mathcal{S}\rvert+\gamma\lvert\mathcal{S}\rvert^{5/2}}{(1-\epsilon)(1-\gamma)^4}\right)$	$\frac{\lvert A\rvert\lvert S\rvert^3(B+\sqrt{\lvert A\rvert})^2}{\delta^4(1-\gamma)^6\lambda^2}\ln\left(\frac{i+1}{p}\right)$	Direct PeMDPs + quadratic-regul. on occupancy
MDRR (Rank et al., 2024)	$\mathcal{O}\left(\frac{\lvert\mathcal{S}\rvert+\gamma\lvert\mathcal{S}\rvert^{5/2}}{(1-\epsilon)(1-\gamma)^4}\right)$	$\frac{\lvert A\rvert\lvert S\rvert^3(B+\sqrt{\lvert A\rvert})^2}{\delta^4(1-\gamma)^6\lambda^2}\ln\left(\frac{i+1}{p}\right)$	Direct PeMDPs + quadratic-regul. on occupancy
PePG (This paper)	$\frac{1-\gamma}{1+2\log(\lvert\mathcal{A}\rvert)}$	$\frac{\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert^2}{\epsilon^2(1-\gamma)^3}$	softmax PeMDPs + entropy regul. on policy
PePG (This paper)	0	$\frac{\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert^2}{\epsilon^2(1-\gamma)^3}$	unregularised softmax PeMDPs

Table 1: Comparison of theoretical performance of SOTA stability-seeking algorithms against PePG.

Implications. (1) We observe that PePG converges to an ϵ-optimal policy in $\frac{|\mathcal{S}||\mathcal{A}|^2}{\epsilon^2(1-\gamma)^3}$ iterations. This reduces the sample complexity required for the existing stability-seeking algorithms by at least an order $\frac{|\mathcal{S}|^2}{(1-\gamma)^3}$, and shows efficiency of using PePG than the algorithms directly optimising the occupancy measures. (2) Additionally, the regularisation parameters needed for the existing algorithms are pretty big and bigger than $\frac{|\mathcal{S}|}{(1-\gamma)^4}$. This is counter-intuitive and does not match the experimental observations. Here, we prove that setting the regularisation parameter to $\frac{1-\gamma}{1+2\log|\mathcal{A}|}$ suffices for proving convergence to optimality. (3) The minimum number of samples required to achieve convergence is proportional to the square of coverage for the softmax PeMDP. This is a ubiquitous quantity dictating convergence of PG-methods in classical RL (Agarwal et al., 2021; Mei et al., 2020), and retraining methods in performative RL (Mandal et al., 2023; Rank et al., 2024).

5 Experimental Analysis

In this section, we empirically compare the performance of PePG in the performative reinforcement learning setting and analyse its behaviour against the state-of-the-art stability-finding methods. [2]

Performative RL Environment. We evaluate PePG in the Gridworld test-bed (Mandal et al., 2023), which has become a standard benchmark in performative RL. This environment consists of a grid where two agents A_1 (the principal) and A_2 (the follower), jointly control an actor navigating from start positions (S) to the goal (G) while avoiding hazards. The environment dynamics are as follows: Agent A_1 proposes a control policy for the actor by selecting one of four directional actions. Agent A_2 can either accept this action (not intervene) or override it with its own directional choice. *This creates a performative environment for A_1, as its effective policy outcomes depend on A_2's responses to its deployed strategy.*

The cost structure follows: visiting blank cells (S) incurs penalty of -0.01, goal cells (F) cost -0.02, hazard cells (H) impose a severe penalty of -0.5, and any intervention by A_2 results in an additional cost of -0.05 for the intervening agent. The response model also follows that of Mandal et al. (2023), i.e., the agent A_2 responds to A_1's policy using a Boltzmann softmax operator. Given A_1's current policy $\boldsymbol{\pi}_1$, we compute the optimal Q-function $Q^{*|\boldsymbol{\pi}_1}$ for each follower agent A_j relative to a perturbed version of the grid world, where each cell types matches A_1's environment with probability 0.7. We then define an average Q-function over the follower agents and determine the collective response policy via Boltzmann softmax $Q^{*|\pi_1}(s,a) = \frac{1}{n}\sum_{j=2}^{n+1} Q_j^{*|\pi_1}(s,a), \pi_2(a|s) = \frac{\exp(\beta\cdot Q^{*|\pi_1}(s,a))}{\sum_{a'}\exp(\beta\cdot Q^{*|\pi_1}(s,a'))}$.

It is important to note that our experimental setup deliberately uses the immediate response model from the original performative RL framework, rather than the gradually shifting environment introduced by Rank et al. (2024) that assumes slow shifts in the environment. Our choice to use the immediate response model presents a more challenging performative setting where the environment responds instantaneously to policy changes. This allows us to demonstrate that unlike MDRR (Rank et al., 2024), PePG can handle the fundamental performative challenge without requiring environmental assumptions that artificially slows down the feedback loop, thereby highlighting the robustness of the proposed PePG approach.

Experimental Setup. We evaluate PePG (with and without entropy regularisation) alongside Mixed Delayed Repeated Retraining (MDRR), which represents the current state-of-the-art in performative reinforcement learning under gradually shifting environments (Rank et al., 2024), and Repeated Policy Optimization with Finite Samples (RPO FS). MDRR has demonstrated significant improvements over traditional repeated retraining methods, by leveraging historical data from multiple deployments, while RPO FS is included as the baseline method from (Mandal et al., 2023) for direct comparison with the original performative RL approach.

[2] Anonymous code repository of PePG implementation is Link. Further ablation studies w.r.t. hyperparameters are in Appendix **??**.

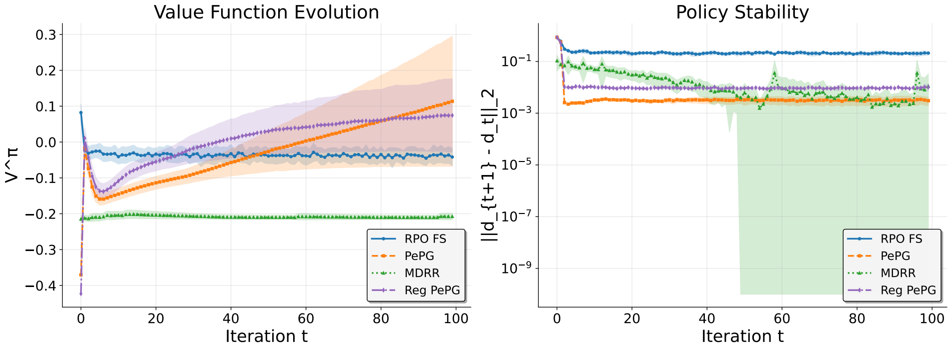

Figure 2: Comparison of evolution in expected average return (both regularised and unregularised) and stability of PePG with SOTA stability-achieving methods. Each algorithm is run for 20 random seeds and 100 iterations.

All experiments use a 8×8 grid with $\gamma = 0.9$, exploration parameter $\epsilon = 0.5$ for initial policy construction, one follower agent A_2, and 100 trajectory samples per iteration. The algorithms share common parameters of $T = 100$ iterations. For regularization, RPO FS and MDRR use $\lambda = 0.1$ from their original experiments, while entropy-regularized PePG uses $\lambda = 2.0$ (ablation studies for this choice are provided in the appendix). PePG uses learning rate $\eta = 0.1$, MDRR employs memory weight $v = 1.1$ for historical data utilization, delayed round parameter $k = 3$, and FTRL parameters $N = B = 10$, while RPO FS follows the finite-sample optimization from Mandal et al.

Results and Observations. Our experimental evaluation across 100 iterations reveals fundamental differences between PePG and MDRR and RPO in the immediate response performative setting. We used shorter training compared to (Rank et al., 2024), as this time-frame sufficiently demonstrates RPO and MDRR's stability convergence and PePG's progression toward optimality.

I. Results: Optimality: The left panel reveals a clear performance hierarchy among the four methods. PePG achieves the highest value function performance, with standard PePG reaching approximately 0.1 and regularized PePG (Reg PePG) reaching 0.05, both showing consistent improvement from initial values around -0.15 and still progressing upward at the end of the 100 iteration window. This steady upward progression highlights PePG's effectiveness in discovering better performative equilibria rather than settling for the first stable solution encountered. RPO FS remains relatively stable around -0.05 throughout training, while MDRR stabilizes at the lowest performance level of approximately -0.2 and remains flat throughout training.

II. Results: Comparison of Optimality- and Stability-seeking Algorithms. The results expose a critical limitation of algorithms designed primarily for stability rather than optimality. MDRR successfully achieves its design goal, with the right panel showing decreasing toward zero in the stability metric $\|d_{t+1} - d_t\|_2$ (the L_2 distance between occupancy measures of consecutive policy iterations), indicating policy stabilization. However, this stability comes at the cost of solution quality, as MDRR becomes trapped in a suboptimal point. The method prioritised finding any stable point over finding an optimal solution. In contrast, both PePG variants exhibit higher policy variability as they actively explore for better solutions. RPO FS maintains moderate stability around 10^{-1} but with limited performance improvement.

6 Discussions, Limitations, and Future Works

We study the problem of Performative Reinforcement learning in tabular MDPs (PeMDPs) using softmax parametrised policies with entropy-regularised objective function, where any action taken by the agent cause potential shift in the MDP's underlying reward and transition dynamics. We are the first to develop PG-type algorithm, PePG, that attains performatively optimality against the existing performative stability-seeking algorithms, affirmatively solving an extended open problem in (Mandal et al., 2023). We also derive the novel performative counterpart of classic Performance Difference Lemma and Policy Gradient Theorem that affirmatively captures this performative nature of the environment we act. We provide a sufficient conditions to prove that PePG converges to an ϵ-ball around performative optimal policy in $\Omega\left(\frac{|\mathcal{S}||\mathcal{A}|^2}{\epsilon^2(1-\gamma)}\right)$ iterations.

As we develop a PG-type algorithm, it will be interesting to see how much can we reduce variance (Wu et al., 2018; Papini et al., 2018) while achieving optimality. We are still in the tabular setting with finite set of state-actions. A potential future direction would be to scale PePG to continuous state-space with large number of state-actions.

REFERENCES

Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *Journal of Machine Learning Research*, 22(98):1–76, 2021. 2, 3, 4, 8

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022. 1

Anas Barakat, John Lazarsfeld, Georgios Piliouras, and Antonios Varvitsiotis. Multi-agent online control with adversarial disturbances. *arXiv preprint arXiv:2506.18814*, 2025. 1

James Bell, Linda Linsefors, Caspar Oesterheld, and Joar Skalse. Reinforcement learning in newcomblike environments. *Advances in Neural Information Processing Systems*, 34:22146–22157, 2021. 2

Allison JB Chaney, Brandon M Stewart, and Barbara E Engelhardt. How algorithmic confounding in recommendation systems increases homogeneity and decreases utility. In *Proceedings of the 12th ACM conference on recommender systems*, pp. 224–232, 2018. 1

Qianyi Chen, Ying Chen, and Bo Li. Practical performative policy learning with strategic agents. *arXiv preprint arXiv:2412.01344*, 2024. 4

Kamil Ciosek and Shimon Whiteson. Expected policy gradients for reinforcement learning. *Journal of Machine Learning Research*, 21(52):1–51, 2020. 3

Itay Eilat and Nir Rosenfeld. Performative recommendation: diversifying content via strategic incentives. In *International Conference on Machine Learning*, pp. 9082–9103. PMLR, 2023. 1

Mohammad Ghavamzadeh and Yaakov Engel. Bayesian policy gradient algorithms. *Advances in neural information processing systems*, 19, 2006. 3

António Góis, Mehrnaz Mofakhami, Fernando P Santos, Gauthier Gidel, and Simon Lacoste-Julien. Performative prediction on games and mechanism design. *arXiv preprint arXiv:2408.05146*, 2024. 1

Moritz Hardt, Meena Jagadeesan, and Celestine Mendler-Dünner. Performative power. *Advances in Neural Information Processing Systems*, 35:22969–22981, 2022. 1

Alexander Havrilla, Yuqing Du, Sharath Chandra Raparthy, Christoforos Nalmpantis, Jane Dwivedi-Yu, Eric Hambro, Sainbayar Sukhbaatar, and Roberta Raileanu. Teaching large language models to reason with reinforcement learning. In *AI for Math Workshop@ ICML 2024*. 1

Michael Herman, Tobias Gindele, Jörg Wagner, Felix Schmitt, and Wolfram Burgard. Inverse reinforcement learning with simultaneous estimation of rewards and dynamics. In *Artificial intelligence and statistics*, pp. 102–110. PMLR, 2016. 4

Zachary Izzo, Lexing Ying, and James Zou. How to learn when data reacts to your model: performative gradient descent. In *International Conference on Machine Learning*, pp. 4641–4650. PMLR, 2021. 1

Zachary Izzo, James Zou, and Lexing Ying. How to learn when data gradually reacts to your model. In *International Conference on Artificial Intelligence and Statistics*, pp. 3998–4035. PMLR, 2022. 1

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the nineteenth international conference on machine learning*, pp. 267–274, 2002. 4

Sham M Kakade. A natural policy gradient. *Advances in neural information processing systems*, 14, 2001. 2

Mahdi Kallel, Debabrota Basu, Riad Akrour, and Carlo D'Eramo. Augmented bayesian policy search. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=OvlcyABNQT. 4

Qiang Li and Hoi-To Wai. State dependent performative prediction with stochastic approximation. In *International Conference on Artificial Intelligence and Statistics*, pp. 3164–3186. PMLR, 2022. 1

Shengshi Li and Lin Yang. Horizon-free learning for markov decision processes and games: stochastically bounded rewards and improved bounds. In *International Conference on Machine Learning*, pp. 20221–20252. PMLR, 2023. 5

Jingbin Liu, Xinyang Gu, and Shuai Liu. Policy optimization reinforcement learning with entropy regularization. *arXiv preprint arXiv:1912.01557*, 2019. 4

Debmalya Mandal and Goran Radanovic. Performative reinforcement learning with linear markov decision process. *arXiv preprint arXiv:2411.05234*, 2024. 2, 3, 4

Debmalya Mandal, Stelios Triantafyllou, and Goran Radanovic. Performative reinforcement learning, 2023. URL https://arxiv.org/abs/2207.00046. 2, 3, 4, 5, 8, 9

Masoud Mansoury, Himan Abdollahpouri, Mykola Pechenizkiy, Bamshad Mobasher, and Robin Burke. Feedback loop and bias amplification in recommender systems. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pp. 2145–2148, 2020. 1

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*, pp. 6820–6829. PMLR, 2020. 3, 4, 5, 8

John P Miller, Juan C Perdomo, and Tijana Zrnic. Outside the echo chamber: Optimizing the performative risk. In *International Conference on Machine Learning*, pp. 7710–7720. PMLR, 2021. 1

Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pp. 1928–1937, 2016. 2

Mehrnaz Mofakhami, Ioannis Mitliagkas, and Gauthier Gidel. Performative prediction with neural networks. In *International Conference on Artificial Intelligence and Statistics*, pp. 11079–11093. PMLR, 2023. 1

Adhyyan Narang, Evan Faulkner, Dmitriy Drusvyatskiy, Maryam Fazel, and Lillian J Ratliff. Multiplayer performative prediction: Learning in decision-dependent games. *Journal of Machine Learning Research*, 24(202):1–56, 2023. 1

Gergely Neu, Anders Jonsson, and Vicenç Gómez. A unified view of entropy-regularized markov decision processes. *arXiv preprint arXiv:1705.07798*, 2017. 4

Matteo Papini, Damiano Binaghi, Giuseppe Canonaco, Matteo Pirotta, and Marcello Restelli. Stochastic variance-reduced policy gradient. In *International conference on machine learning*, pp. 4026–4035. PMLR, 2018. 9

Juan Perdomo, Tijana Zrnic, Celestine Mendler-Dünner, and Moritz Hardt. Performative prediction. In *International Conference on Machine Learning*, pp. 7599–7609. PMLR, 2020. 1

Georgios Piliouras and Fang-Yi Yu. Multi-agent performative prediction: From global stability and optimality to chaos. In *Proceedings of the 24th ACM Conference on Economics and Computation*, pp. 1047–1074, 2023. 1

Vasilis Pollatos, Debmalya Mandal, and Goran Radanovic. On corruption-robustness in performative reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 19939–19947, 2025. 4

Ben Rank, Stelios Triantafyllou, Debmalya Mandal, and Goran Radanovic. Performative reinforcement learning in gradually shifting environments. *arXiv preprint arXiv:2402.09838*, 2024. 2, 4, 5, 8, 9

John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pp. 1889–1897, 2015. 2

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. In *arXiv preprint arXiv:1707.06347*, 2017. 2

Wenjie Shi, Shiji Song, and Cheng Wu. Soft policy gradient method for maximum entropy deep reinforcement learning. *arXiv preprint arXiv:1909.03198*, 2019. 4

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *International conference on machine learning*, pp. 387–395. Pmlr, 2014. 1, 2, 4, 5

Jason St. John, Christian Herwig, Diana Kafkes, Jovan Mitrevski, William A Pellico, Gabriel N Perdue, Andres Quintero-Parra, Brian A Schupbach, Kiyomi Seiya, Nhan Tran, et al. Real-time artificial intelligence for accelerator control: A study at the fermilab booster. *Physical Review Accelerators and Beams*, 24(10):104601, 2021. 1

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press, 1998. 1

Richard S. Sutton, David A. McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems 12 (NIPS 1999)*, pp. 1057–1063. The MIT Press, 1999. 2, 3, 5

Jiexin Wang and Eiji Uchibe. Reward-punishment reinforcement learning with maximum entropy. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7. IEEE, 2024. 4

Yue Wang and Shaofeng Zou. Policy gradient method for robust reinforcement learning. In *International conference on machine learning*, pp. 23484–23526. PMLR, 2022. 3

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992. 2, 3, 5

Cathy Wu, Aravind Rajeswaran, Yan Duan, Vikash Kumar, Alexandre M Bayen, Sham Kakade, Igor Mordatch, and Pieter Abbeel. Variance reduction for policy gradient with action-dependent factorized baselines. *arXiv preprint arXiv:1803.07246*, 2018. 9

Rui Yuan, Robert M Gower, and Alessandro Lazaric. A general sample complexity analysis of vanilla policy gradient. In *International Conference on Artificial Intelligence and Statistics*, pp. 3332–3380. PMLR, 2022. 2, 3

Angela Zhang, Lei Xing, James Zou, and Joseph C Wu. Shifting machine learning for healthcare from development to deployment and from models to data. *Nature biomedical engineering*, 6(12):1330–1345, 2022. 1

Rui Zhao, Xudong Sun, and Volker Tresp. Maximum entropy-regularized multi-goal reinforcement learning. In *International Conference on Machine Learning*, pp. 7553–7562. PMLR, 2019. 4