BEYOND STATIONARITY: CONVERGENCE ANALYSIS OF STOCHASTIC SOFTMAX POLICY GRADIENT METHODS

Sara Klein, Simon Weissmann & Leif Döring
Institute of Mathematics, University of Mannheim
{sara.klein, simon.weissmann, leif.doering}@uni-mannheim.de

ABSTRACT

Markov Decision Processes (MDPs) are a formal framework for modeling and solving sequential decision-making problems. In finite-time horizons such problems are relevant for instance for optimal stopping or specific supply chain problems, but also in the training of large language models. In contrast to infinite horizon MDPs optimal policies are not stationary, policies must be learned for every single epoch. In practice all parameters are often trained simultaneously, ignoring the inherent structure suggested by dynamic programming. This paper introduces a combination of dynamic programming and policy gradient called dynamic policy gradient, where the parameters are trained backwards in time.

For the tabular softmax parametrisation we carry out the convergence analysis for simultaneous and dynamic policy gradient towards global optima, both in the exact and sampled gradient settings without regularisation. It turns out that the use of dynamic policy gradient training much better exploits the structure of finite-time problems which is reflected in improved convergence bounds.

1 INTRODUCTION

Policy gradient (PG) methods continue to enjoy great popularity in practice due to their model-free nature and high flexibility. Despite their far-reaching history (Williams, 1992; Sutton et al., 1999; Konda & Tsitsiklis, 1999; Kakade, 2001), there were no proofs for the global convergence of these algorithms for a long time. Nevertheless, they have been very successful in many applications, which is why numerous variants have been developed in the last few decades, whose convergence analysis, if available, was mostly limited to convergence to stationary points (Pirotta et al., 2013; Schulman et al., 2015; Papini et al., 2018; Clavera et al., 2018; Shen et al., 2019; Xu et al., 2020b; Huang et al., 2020; Xu et al., 2020a; Huang et al., 2022). In recent years, notable advancements have been achieved in the convergence analysis towards global optima (Fazel et al., 2018; Agarwal et al., 2021; Mei et al., 2020; Bhandari & Russo, 2021; 2022; Cen et al., 2022; Xiao, 2022; Yuan et al., 2022; Alfano & Rebeschini, 2023; Johnson et al., 2023). These achievements are partially attributed to the utilisation of (weak) gradient domination or Polyak-Łojasiewicz (PL) inequalities (lower bounds on the gradient) (Polyak, 1963).

As examined in Karimi et al. (2016) a PL-inequality and β-smoothness (i.e. β-Lipschitz continuity of the gradient) implies a linear convergence rate for gradient descent methods. In certain cases, only a weaker form of the PL inequality can be derived, which states that it is only possible to lower bound the norm of the gradient instead of the squared norm of the gradient by the distance to the optimum. Despite this limitation, $\mathcal{O}(1/n)$-convergence can still be achieved in some instances.

This article deals with PG algorithms for finite-time MDPs. Finite-time MDPs differ from discounted infinite-time MDPs in that the optimal policies are not stationary, i.e. depend on the epochs. While a lot of recent theoretical research focused on discounted MDPs with infinite-time horizon not much is known for finite-time MDPs. However, there are many relevant real world applications which require non-stationary finite-time solutions such as inventory management in hospital supply chains (Abu Zwaida et al., 2021) or optimal stopping in finance (Becker et al., 2019). There is a prevailing thought that finite-time MDPs do not require additional scrutiny as they can be transformed

into infinite horizon MDPs by adding an additional time-coordinate. Seeing finite-time MDPs this way leads to a training procedure in which parameters for all epochs are trained simultaneously, see for instance Guin & Bhatnagar (2023). While there are practical reasons to go that way, we will see below that ignoring the structure of the problem yields worse convergence bounds. The aim of this article is two-fold. Firstly, we analyse the simultaneous PG algorithm. The analysis for exact gradients goes along arguments of recent articles, the analysis of the stochastic PG case is novel. Secondly, we introduce a new approach to PG for finite-time MDPs. We exploit the dynamic programming structure and view the MDP as a nested sequence of contextual bandits. Essentially, our algorithm performs a sequence of PG algorithms backwards in time with carefully chosen epoch dependent training steps. We compare the exact and stochastic analysis to the simultaneous approach. Dynamic PG can bee seen as a concrete algorithm for Policy Search by Dynamic Programming, where policy gradient is used to solve the one-step MDP (Bagnell et al., 2003; Scherrer, 2014). There are some recent articles also studying PG of finite-time horizon MDPs from a different perspective considering fictitious discount algorithms (Guo et al., 2022) or finite-time linear quadratic control problems (Hambly et al., 2021; 2023; Zhang et al., 2021; 2023b;a; Zhang & Başar, 2023).

This article can be seen to extend a series of recent articles from discounted MDPs to finite-time MDPs. In Agarwal et al. (2021), the global asymptotic convergence of PG is demonstrated under tabular softmax parametrisation, and convergence rates are derived using log-barrier regularisation and natural policy gradient. Building upon this work, Mei et al. (2020) showed the first convergence rates for PG using non-uniform PL-inequalities (Mei et al., 2021), specifically for tabular softmax parametrisation. The convergence rate relies heavily on the discount factor as $(1-\gamma)^{-6}$ and does not not readily convert to non-discounted MDPs. Through careful analysis, we establish upper bounds involving H^5 for simultaneous PG, contrasting with H^3 for dynamic PG. Essentially, dynamic PG offers a clear advantage. Examining the PG theorem for finite-time MDPs reveals that early epochs should be trained less if policies for later epochs are suboptimal. A badly learned Q-function-to-go leads to badly directed gradients in early epochs. Thus, simultaneous training yields ineffective early epoch training, addressed by our dynamic PG algorithm, optimizing policies backward in time with more training steps. To illustrate this phenomenon we implemented a simple toy example where the advantage of dynamic PG becomes visible. In Figure 1 one can see 5 simulations of the dynamic PG with different target accuracies (blue curves) plotted against one version of the simultaneous PG with target accuracy 0.1 (dashed magenta curve). The time-horizon is chosen as $H = 5$. More details on the example can be found in Appendix E.

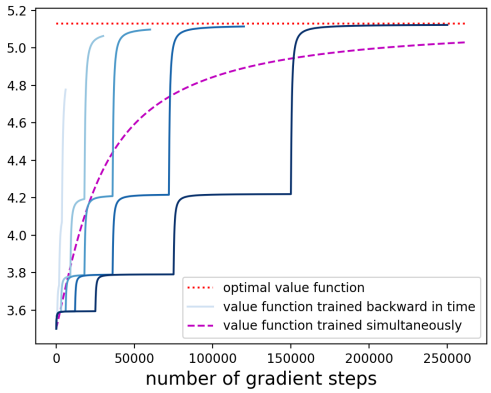

Figure 1: Evolution of the value function during training.

A main further contribution of this article is a stochastic analysis, where we abandon the assumption that the exact gradient is known and focus on the model free stochastic PG method. For this type of algorithm, very little is known about convergence to global optima even in the discounted case. Many recent articles consider variants such as natural PG or mirror descent to analyse the stochastic scenario (Agarwal et al., 2021; Fatkhullin et al., 2023; Xiao, 2022; Alfano et al., 2023). Ding et al. (2022) derive complexity bounds for entropy-regularised stochastic PG. They use a well-chosen stopping time which measures the distance to the set of optimal parameters, and simultaneously guarantees convergence to the regularised optimum prior to the occurrence of the stopping time by using a small enough step size and large enough batch size. As we are interested in convergence to the unregularised optimum, we consider stochastic softmax PG without regularisation. Similar to the previous idea, we construct a different stopping time, which allows us to derive complexity bounds for an approximation arbitrarily close to the global optimum that does not require a set of optimal parameters and this is relevant when considering softmax parametrisation. To the best of our knowledge, the results presented in this paper provide the first convergence analysis for dynamic programming inspired PG under softmax parametrisation in the finite-time MDP setting. Both for exact and batch sampled policy gradients without regularisation.

2 FINITE-TIME HORIZON MDPS AND POLICY GRADIENT METHODS.

A finite-time MDP is defined by a tuple $(\mathcal{H}, \mathcal{S}, \mathcal{A}, r, p)$ with $\mathcal{H} = \{0, \dots, H-1\}$ decision epochs, finite state space $\mathcal{S} = \mathcal{S}_0 \cup \dots \cup \mathcal{S}_{H-1}$, finite action space $\mathcal{A} = \bigcup_{s \in \mathcal{S}} \mathcal{A}_s$, a reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and transition function $p : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ with $p(\mathcal{S}_{h+1}|s,a) = 1$ for every $h < H-1$, $s \in \mathcal{S}_h$ and $a \in \mathcal{A}_s$. Here $\Delta(D)$ denotes the set of all probability measures over a finite set D.

Throughout the article $\pi = (\pi_h)_{h=0}^{H-1}$ denotes a time-dependent policy, where $\pi_h : \mathcal{S}_h \to \Delta(\mathcal{A})$ is the policy in decision epoch $h \in \mathcal{H}$ with $\pi_h(\mathcal{A}_s|s) = 1$ for every $s \in \mathcal{S}_h$. It is well-known that in contrast to discounted infinite-time horizon MDPs non-stationary policies are needed to optimise finite-time MDPs. An optimal policy in time point h depends on the time horizon until the end of the problem (see for example Puterman (2005)). The epoch-dependent value functions under policy π are defined by

$$V_h^{\pi_{(h)}}(\mu_h) := \mathbb{E}_{\mu_h}^{\pi_{(h)}}\Big[\sum_{k=h}^{H-1} r(S_k, A_k)\Big], \quad h \in \mathcal{H}, \tag{1}$$

where μ_h is an initial distribution, $\pi_{(h)} = (\pi_k)_{k=h}^{H-1}$ denotes the sub-policy of π from h to $H-1$ and $\mathbb{E}_{\mu_h}^{\pi_{(h)}}$ is the expectation under the measure such that $S_h \sim \mu_h$, $A_k \sim \pi_k(\cdot|S_k)$ and $S_{k+1} \sim p(\cdot|S_k, A_k)$ for $h \leq k < H-1$. The target is to find a (time-dependent) policy that maximises the state-value function V_0 at time 0. In the following we will discuss two approaches to solve finite-time MDPs with PG:

- An algorithm that is often used in practice, where parametrised policies are trained simultaneously, i.e. the parameters for $\pi_0, ..., \pi_{H-1}$ are trained at once using the objective V_0.
- A new algorithm that trains the parameters sequentially starting at the last epoch. We call this scheme dynamic PG because it combines dynamic programming (backwards induction) and PG.

In fact, one can also consider PG algorithms that train stationary policies (i.e. independent of h) for finite-time MDPs. However, this violates the intrinsic nature of finite-time MDPs (optimal policies will only be stationary in trivial cases). In order to carry out a complete theoretical analysis assumptions are required. In this article we will assume that all policies are softmax parametrised, an assumption that appeared frequently in the past years. It is a first step towards a full understanding and already indicates why PG methods should use the dynamic programming structure inherent in finite-time MDPs. This paper should not be seen as limited to the softmax case, but more like a kick-off to analyse a new approach which is beneficial in many scenarios.

Simultaneous Policy Gradient. Let us start by formulating the simultaneous PG algorithm that is often used in practice. The action spaces may depend on the current state and the numbers of possible actions in epoch h is denoted by $d_h = \sum_{s \in \mathcal{S}_h} |\mathcal{A}_s|$. To perform a PG algorithm all policies π_h (or the entire policy π) must be parametrised. While the algorithm does not require a particular policy we will analyse the tabular softmax parametrisation

$$\pi^\theta(a|s_h) = \frac{\exp(\theta(s_h, a))}{\sum_{a'} \exp(\theta(s_h, a'))}, \quad \theta = (\theta(s_h, a))_{s_h \in \mathcal{S}^{[\mathcal{H}]}, a \in \mathcal{A}_{s_h}} \in \mathbb{R}^{\sum_h d_h}, \tag{2}$$

where the notation $\mathcal{S}^{[\mathcal{H}]}$ defines the enlarged state space, containing all possible states associated to their epoch (see Remark A.1 for more details). The tabular softmax parametrisation uses a single parameter for each possible state-action pair at all epochs. Other parametrised policies, e.g. neural networks, take states from all epochs, i.e. from the enlarged state space $\mathcal{S}^{[\mathcal{H}]}$, as input variables. The simultaneous PG algorithm trains all parameters at once and solves the optimisation problem (to maximize the state value function at time 0) by gradient ascent over all parameters (all epochs) simultaneously.

Most importantly, the algorithm does not treat epochs differently, the same training effort goes into all epochs. For later use the objective function will be denoted by

$$J(\theta, \mu) := V_0^{\pi^\theta}(\mu) = \mathbb{E}_\mu^{\pi^\theta}\Big[\sum_{h=0}^{H-1} r(S_h, A_h)\Big] \tag{3}$$

Algorithm 1: Simultaneous Policy Gradient for finite-time MDPs

Result: Approximate policy $\hat{\pi}^* \approx \pi^*$
initialise $\theta^{(0)} \in \mathbb{R}^{\sum_h d_h}$
Choose fixed step sizes $\eta > 0$, number of training steps N and start distribution μ
for $n = 0, \dots, N-1$ **do**
  $\theta^{(n+1)} = \theta^{(n)} + \eta \nabla_\theta V_0^{\pi^{\theta^{(n)}}}(\mu)\big|_{\theta^{(n)}}$
end
Set $\hat{\pi}^* = \pi^{\theta^{(N)}}$

Furthermore, let $\rho_\mu^{\pi^\theta}(s) = \sum_{h=0}^{H-1} \mathbb{P}_\mu^{\pi^\theta}(S_h = s)$ be the state-visitation measure on $\mathcal{S}$ and $d_\mu^{\pi^\theta}(s) = \frac{1}{H}\rho_\mu^{\pi^\theta}(s)$ be the normalised state-visitation distribution. We denote by $J^*(\mu) = \sup_\theta J(\theta, \mu)$ the optimal value of the objective function and note that $J^*(\mu) = V_0^*(\mu) = \sup_{\pi:\text{ Policy}} V_0^\pi(\mu)$ under the tabular softmax parametrisation, as an optimal policy can be approximated arbitrarily well.

Dynamic Policy Gradient. First of all, recall that the inherent structure of finite-time MDPs is a backwards induction principle (dynamic programming), see for instance (Puterman, 2005). To see backwards induction used in learning algorithms we refer for instance to Bertsekas & Tsitsiklis (1996, Sec 6.5). In a way, finite-time MDPs can be viewed as nested contextual bandits. The dynamic PG approach suggested in this article builds upon this intrinsic structure and sets on top a PG scheme. Consider H different parameters $\theta_0, \dots, \theta_{H-1}$ such that $\theta_h \in \mathbb{R}^{d_h}$. A parametric policy $(\pi^{\theta_h})_{h=0}^{H-1}$ is defined such that the policy in epoch h depends only on the parameter θ_h. An example is the tabular softmax parametrisation formulated slightly differently than above. For each decision epoch $h \in \mathcal{H}$ the tabular softmax parametrisation is given by

$$\pi^{\theta_h}(a|s) = \frac{\exp(\theta_h(s,a))}{\sum_{a' \in \mathcal{A}} \exp(\theta_h(s,a'))}, \quad \theta_h = (\theta_h(s,a))_{s \in \mathcal{S}_h, a \in \mathcal{A}_s} \in \mathbb{R}^{d_h}. \tag{4}$$

The total dimension of the parameter tensor $(\theta_0, \dots, \theta_{H-1})$ equals the one of θ from the Equation 2 because $\theta_h(s_h, a) = \theta(s_h, a)$ for $s_h \in \mathcal{S}_h \subset \mathcal{S}^{[\mathcal{H}]}$. The difference is that the epoch dependence is made more explicit in Equation 4.

The main idea of this approach is as follows. The dynamic programming perspective suggests to learn policies backwards in time. Thus, we start by training the last parameter vector θ_{H-1} on the sub-problem V_{H-1}, a one-step MDP which can be viewed as contextual bandit. After convergence up to some termination condition, it is known how to act near optimality in the last epoch and one can proceed to train the parameter vector from previous epochs by exploiting the knowledge of acting near optimal in the future. This is what the proposed dynamic PG algorithm does. A policy is trained up to some termination condition and then used to optimise an epoch earlier.

Algorithm 2: Dynamic Policy Gradient for finite-time MDPs

Result: Approximate policy $\hat{\pi}^* \approx \pi^*$
initialise $\theta^{(0)} = (\theta_0^{(0)}, \dots, \theta_{H-1}^{(0)}) \in \Theta$
for $h = H-1, \dots, 0$ **do**
  Choose fixed step size η_h, number of training steps N_h and start distribution μ_h
  for $n = 0, \dots, N_h - 1$ **do**
    $\theta_h^{(n+1)} = \theta_h^{(n)} + \eta_h \nabla_{\theta_h} V_h^{(\pi^{\theta_h}, \hat{\pi}^*_{(h+1)})}(\mu_h)\big|_{\theta_h^{(n)}}$
  end
  Set $\hat{\pi}_h^* = \pi^{\theta_h^{(N_h)}}$
end

A bit of notation is needed to analyse this approach. Given any fixed policy $\tilde{\pi}$, the objective function J_h in epoch h is defined to be the h-state value function in state under the extended policy

$(\pi^{\theta_h}, \tilde{\pi}_{(h+1)}) := (\pi^{\theta_h}, \tilde{\pi}_{h+1}, \dots, \tilde{\pi}_{H-1})$,

$$J_h(\theta_h, \tilde{\pi}_{(h+1)}, \mu_h) := V_h^{(\pi^{\theta_h}, \tilde{\pi}_{(h+1)})}(\mu_h) = \mathbb{E}_{\mu_h}^{(\pi^{\theta_h}, \tilde{\pi}_{(h+1)})}\Big[\sum_{k=h}^{H-1} r(S_k, A_k)\Big]. \tag{5}$$

While the notation is a bit heavy the intuition behind is easy to understand. If the policy after epoch h is already trained (this is $\tilde{\pi}_{(h+1)}$) then J_h as a function of θ_h is the parametrised dependence of the value function when only the policy for epoch h is changed. Gradient ascent is then used to find a parameter θ_h^* that maximises $J_h(\cdot, \tilde{\pi}_{(h+1)}, \delta_s)$, for all $s \in \mathcal{S}_h$, where δ_s the dirac measure on s. Note that to train θ_h one chooses $\tilde{\pi}_{(h+1)} = \hat{\pi}^*_{(h+1)}$ in Algorithm 2.

A priori it is not clear if simultaneous or dynamic programming inspired training is more efficient. Dynamic PG has an additional loop but trains less parameters at once. We give a detailed analysis for the tabular softmax parametrisation but want to give a heuristic argument why simultaneous training is not favorable. The policy gradient theorem, see Theorem A.5, states that

$$\nabla J(\theta, \mu) = \sum_{s_h \in \mathcal{S}^{[\mathcal{H}]}} \tilde{\rho}_\mu^{\pi^\theta}(s_h) \sum_{a \in \mathcal{A}_{s_h}} \pi^\theta(a|s_h) \nabla \log(\pi^\theta(a|s_h)) Q_h^{\pi^\theta}(s_h, a),$$

involving Q-values under the current policy[1]. It implies that training policies at earlier epochs are massively influenced by estimation errors of $Q_h^{\pi^\theta}$. Reasonable training of optimal decisions is only possible if all later epochs have been trained well, i.e. $Q_h^{\pi^\theta} \approx Q_h^*$. This may lead to inefficiency in earlier epochs when training all epochs simultaneously. It is important to note that the policy gradient formula is independent of the parametrisation. While our precise analysis is only carried out for tabular softmax parametrisations this general heuristic remains valid for all classes of policies.

Assumption 2.1. Throughout the remaining manuscript we assume that the rewards are bounded in $[0, R^*]$, for some $R^* > 0$. The positivity is no restriction of generality, bounded negative rewards can be shifted using the base-line trick.

In what follows we will always assume the tabular softmax parametrisation and analyse both PG schemes. First under the assumption of exact gradients, then with sampled gradients à la REINFORCE.

3 Convergence of Softmax Policy Gradient with exact gradients

In the following, we analyse the convergence behavior of the simultaneous as well as the dynamic approach under the assumption to have access to exact gradient computation. The presented convergence analysis in both settings is inspired from the discounted setting considered recently in Agarwal et al. (2021); Mei et al. (2020). The idea is to combine smoothness of the objective function and a (weak) PL-inequality in order to derive a global convergence result.

3.1 Simultaneous Policy Gradient

To prove convergence in the simultaneous approach we will interpret the finite-time MDP as an undiscounted stationary problem with state-space $\mathcal{S}^{[H]}$ and deterministic absorption time H. This MDP is undiscounted but terminates in finite-time. Building upon Agarwal et al. (2021); Mei et al. (2020); Yuan et al. (2022) we prove that the objective function defined in Equation 3 is β-smooth with parameter $\beta = H^2 R^*(2 - \frac{1}{|\mathcal{A}|})$ and satisfies a weak PL-inequality of the form

$$\|\nabla J(\theta, \mu)\|_2 \geq \frac{\min_{s_h \in \mathcal{S}^{[\mathcal{H}]}} \pi^\theta(a^*(s_h)|s_h)}{\sqrt{|\mathcal{S}^{[\mathcal{H}]}|}} \Big\| \frac{d_\mu^{\pi^*}}{d_\mu^{\pi^\theta}} \Big\|_\infty^{-1} (J^*(\mu) - J(\theta, \mu)).$$

Here π^* denotes a fixed but arbitrary deterministic optimal policy for the enlarged state space $\mathcal{S}^{[\mathcal{H}]}$ and $a^*(s_h) = \text{argmax}_{a \in \mathcal{A}_{s_h}} \pi^*(a|s_h)$ is the best action in state s_h. The term

$$\Big\| \frac{d_\mu^{\pi^*}}{d_\mu^{\pi^\theta}} \Big\|_\infty := \max_{s \in \mathcal{S}} \frac{d_\mu^{\pi^*}(s)}{d_\mu^{\pi^\theta}(s)} \tag{6}$$

[1] See Appendix A, Equation 12 and Equation 13 for the definition of the state-action value function Q and the enlarged state visitation measure $\tilde{\rho}$.

is the distribution mismatch coefficient introduced in Agarwal et al. (2021, Def 3.1). Both properties are shown in Appendix B.1. To ensure that the distribution mismatch coefficient can be bounded from below uniformly in θ (see also Remark B.4) we make the following assumption.

Assumption 3.1. For the simultaneous PG algorithm we assume that the state space is constant over all epochs, i.e. $\mathcal{S}_h = \mathcal{S}$ for all epochs.

As already pointed out in Mei et al. (2020) one key challenge in providing global convergence is to bound the term $\min_{s\in\mathcal{S}} \pi^\theta(a_h^*(s)|s)$ from below uniformly in θ appearing in the gradient ascent updates. Techniques introduced in Agarwal et al. (2021) can be extended to the finite-horizon setting to prove asymptotic convergence towards global optima. This can then be used to bound $c = c(\theta^{(0)}) = \inf_n \min_{s\in\mathcal{S}} \pi^{\theta^{(n)}}(a_h^*(s)|s) > 0$ (Lemma B.5). Combining smoothness and the gradient domination property results in the following global convergence result.

Theorem 3.2. *Under Assumption 3.1, let μ be a probability measure such that $\mu(s) > 0$ for all $s \in \mathcal{S}$, let $\eta = \frac{1}{5H^2R^*}$ and consider the sequence $(\theta^{(n)})$ generated by Algorithm 1 with arbitrary initialisation $\theta^{(0)}$. For $\epsilon > 0$ choose the number of training steps as $N = \frac{10H^5R^*|\mathcal{S}|}{c^2\epsilon}\left\|\frac{d_\mu^{\pi^*}}{\mu}\right\|_\infty^2$. Then it holds that*

$$V_0^*(\mu) - V_0^{\pi^{\theta^{(N)}}}(\mu) \leq \epsilon.$$

One can compare this result to Mei et al. (2020, Thm 4) for discounted MDPs. A discounted MDP can be seen as an undiscounted MDP stopped at an independent geometric random variable with mean $(1-\gamma)^{-1}$. Thus, it comes as no surprise that algorithms with deterministic absorption time H have analogous estimates with H instead of $(1-\gamma)^{-1}$. See Remark B.6 for a detailed comparison. Furthermore, it is noteworthy that it cannot be proven that c is independent of H. We omitted this dependency when we compare to the discounted case because the model dependent constant there could also depend on γ in the same sense.

3.2 Dynamic Policy Gradient

We now come to the first main contribution of this work, an improved bound for the convergence of the dynamic PG algorithm. The optimisation objectives are J_h defined in Equation 5. The structure of proving convergence is as follows. For each fixed $h \in \mathcal{H}$ we provide global convergence given that the policy after h is fixed and denoted by $\tilde{\pi}$. After having established bounds for each decision epoch, we apply backwards induction to derive complexity bounds on the total error accumulated over all decision epochs. The β-smoothness for different J_h is then reflected in different training steps for different epochs.

The backwards induction setting can be described as a nested sequence of contextual bandits (one-step MDPs) and thus, can be analysed using results from the discounted setting by choosing $\gamma = 0$. Using PG estimates for dicounted MDPs (Mei et al., 2020; Yuan et al., 2022) we prove in Appendix B.2 that the objective J_h from Equation 5 is a smooth function in θ_h with parameter $\beta_h = 2(H-h)R^*$ and satisfies also a weak PL-inequality of the form

$$\|\nabla J_h(\theta_h, \tilde{\pi}_{(h+1)}, \mu_h)\|_2 \geq \min_{s\in\mathcal{S}_h} \pi^{\theta_h}(a_h^*(s)|s)(J_h^*(\tilde{\pi}_{(h+1)}, \mu_h) - J_h(\theta_h, \tilde{\pi}_{(h+1)}, \mu_h)).$$

It is crucial to keep in mind that classical theory from non-convex optimisation tells us that less smooth (large β) functions must be trained with more gradient steps. It becomes clear that the dynamic PG algorithm should spend less training effort on later epochs (earlier in the algorithm) and more training effort on earlier epochs (later in the algorithm). In fact, we make use of this observation by applying backwards induction in order to improve the convergence behavior depending on H (see Theorem 4.2). The main challenge is again to bound $\min_{s\in\mathcal{S}} \pi^{\theta_h}(a_h^*(s)|s)$ from below uniformly in θ_h appearing in the gradient ascent updates from Algorithm 2. In this setting the required asymptotic convergence follows directly from the one-step MDP viewpoint using $\gamma = 0$ obtained in Agarwal et al. (2021, Thm 5) and it holds $c_h = \inf_{n\geq 0} \min_{s\in\mathcal{S}_h} \pi^{\theta_h^{(n)}}(a_h^*(s)|s) > 0$ (Lemma B.10).

There is another subtle advantage in the backwards induction point of view. The contextual bandit interpretation allows using refinements of estimates for the special case of contextual bandits. A slight generalisation of work of Mei et al. (2020) for stochastic bandits shows that the unpleasant unknown constants c_h simplify if the PG algorithm is uniformly initialised:

Proposition 3.3. *For fixed $h \in \mathcal{H}$, let μ_h be a probability measure such that $\mu_h(s) > 0$ for all $s \in \mathcal{S}_h$ and let $0 < \eta_h \leq \frac{1}{2(H-h)R^*}$. Let $\theta_h^{(0)} \in \mathcal{R}^{d_h}$ be an initialisation such that the initial policy is a uniform distribution, then $c_h = \frac{1}{|\mathcal{A}|} > 0$.*

This property is in sharp contrast to the simultaneous approach, where to the best of our knowledge it is not known how to lower bound c explicitly. Comparing the proofs of $c > 0$ and $c_h > 0$ one can see that this advantage comes from the backward inductive approach and is due to fixed future policies which are not changing during training. For fixed decision epoch h combining β-smoothness and weak PL inequality yields the following global convergence result for the dynamic PG generated in Algorithm 2.

Lemma 3.4. *For fixed $h \in \mathcal{H}$, let μ_h be a probability measure such that $\mu_h(s) > 0$ for all $s \in \mathcal{S}_h$, let $\eta_h = \frac{1}{2(H-h)R^*}$ and consider the sequence $(\theta_h^{(n)})$ generated by Algorithm 2 with arbitrary initialisation $\theta_h^{(0)}$ and $\tilde{\pi}$. For $\epsilon > 0$ choose the number of training steps as $N_h = \frac{4(H-h)R^*}{c_h^2 \epsilon}$. Then it holds that*

$$V_h^{(\pi_h^*, \tilde{\pi}_{(h+1)})}(\mu_h) - V_h^{(\pi^{\theta_h^{(N_h)}}, \tilde{\pi}_{(h+1)})}(\mu_h) \leq \epsilon$$

Moreover, if $\theta_h^{(0)}$ initialises the uniform distribution the constants c_h can be replaced by $\frac{1}{|\mathcal{A}|}$.

The error bound depends on the time horizon up to the last time point, meaning intuitively that an optimal policy for earlier time points in the MDP (smaller h) is harder to achieve and requires a longer learning period then later time points (h near to H). We remark that the assumption on μ_h is not a sharp restriction and can be achieved by using a strictly positive start distribution μ on $\mathcal{S}_0$ followed by a uniformly distributed policy. Note that assuming a positive start distribution is common in the literature and Mei et al. (2020) showed the necessity of this assumption. Accumulating errors over time we can now derive the analogous estimates to the simultaneous PG approach. We obtain a linear accumulation such that an $\frac{\epsilon}{H}$-error in each time point h results in an overall error of ϵ which appears naturally from the dynamic programming structure of the algorithm.

Theorem 3.5. *For all $h \in \mathcal{H}$, let μ_h be probability measures such that $\mu_h(s) > 0$ for all $s \in \mathcal{S}_h$, let $\eta_h = \frac{1}{2(H-h)R^*}$. For $\epsilon > 0$ choose the number of training steps as $N_h = \frac{4(H-h)HR^*}{c_h^2 \epsilon} \left\| \frac{1}{\mu_h} \right\|_\infty$. Then for the final policy from Algorithm 2, $\hat{\pi}^* = (\pi^{\theta_0^{(N_0)}}, \dots, \pi^{\theta_{H-1}^{(N_{H-1})}})$, it holds for all $s \in \mathcal{S}_0$ that*

$$V_0^*(s) - V_0^{\hat{\pi}^*}(s) \leq \epsilon.$$

If $\theta_h^{(0)}$ initialises the uniform distribution the constants c_h can be replaced by $\frac{1}{|\mathcal{A}|}$.

3.3 Comparison of the algorithms

Comparing Theorem 3.5 to the convergence rate for simultaneous PG in Theorem 3.2, we first highlight that the constant c_h in the dynamic approach can be explicitly computed under uniform initialisation. This has not yet been established in the simultaneous PG (see Remark B.11) and especially it cannot be guaranteed that c is independent of the time horizon. Second, we compare the overall dependence of the training steps on the time horizon. In the dynamic approach $\sum_h N_h$ scales with H^3 in comparison to H^5 in the convergence rate for the simultaneous approach. In particular for large time horizons the theoretical analysis shows that reaching a given accuracy is more costly for simultaneous training of parameters. In the dynamic PG the powers are due to the smoothness constant, the $\frac{\epsilon}{H}$ error which we have to achieve in every epoch and finally the sum over all epochs. In comparison, in the simultaneous PG a power of 2 is due to the smoothness constant, another power of 2 is due to the distribution mismatch coefficient in the PL-inequality which we need to bound uniformly in θ (see also Remark B.3) and the last power is due to the enlarged state space $|\mathcal{S}^{[H]}| = |\mathcal{S}|H$. Note that we just compare upper bounds. However, we refer to Appendix E for a toy example visualising that the rate of convergence in both approaches is of order $\mathcal{O}(\frac{1}{n})$ and the constants in the dynamic approach are indeed better then for the simultaneous approach.

4 CONVERGENCE ANALYSIS OF STOCHASTIC SOFTMAX POLICY GRADIENT

In the previous section, we have derived global convergence guarantees for solving a finite-time MDP via simultaneous as well as dynamic PG with exact gradient computation. However, in practical scenarios assuming access to exact gradients is not feasible, since the transition function p of the underlying MDP is unknown. In the following section, we want to relax this assumption by replacing the exact gradient by a stochastic approximation. To be more precise, we view a model-free setting where we are only able to generate trajectories of the finite-time MDP. These trajectories are used to formulate the stochastic PG method for training the parameters in both the simultaneous and dynamic approach.

Although in both approaches we are able to guarantee almost sure asymptotic convergence similar to the exact PG scheme, we are no longer able to control the constants c and c_h respectively along trajectories of the stochastic PG scheme due to the randomness in our iterations. Therefore, the derived lower bound in the weak PL-inequality may degenerate in general. In order to derive complexity bounds in the stochastic scenario, we make use of the crucial property that c (and c_h respectively) remain strictly positive along the trajectory of the exact PG scheme. To do so, we introduce the stopping times τ and τ_h stopping the scheme when the stochastic PG trajectory is too far away from the exact PG trajectory (under same initialisation). Hence, conditioning on $\{\tau \geq n\}$ (and $\{\tau_h \geq n\}$ respectively) forces the stochastic PG to remain close to the exact PG scheme and hence, guarantees non-degenerated weak PL-inequalities. The proof structure in the stochastic setting is then two-fold:

1. We derive a rate of convergence of the stochastic PG scheme under non-degenerated weak PL-inequality on the event $\{\tau \geq n\}$. Since we consider a constant step size, the batch size needs to be increased sufficiently fast for controlling the variance occurring through the stochastic approximation scheme. See Lemma D.4 and Lemma D.8.
2. We introduce a second rule for increasing the batch-size depending on a tolerance $\delta > 0$ leading to $\mathbb{P}(\tau \leq n) < \delta$. This means, that one forces the stochastic PG to remain close to the exact PG with high probability. See Lemma D.5 and Lemma D.9.

A similar proof strategy has been introduced in Ding et al. (2022) for proving convergence for entropy-regularised stochastic PG in the discounted case. Their analysis heavily depends on the existence of an optimal parameter which is due to regularisation. In the unregularised problem this is not the case since the softmax parameters usually diverge to $+/-\infty$ in order to approximate a deterministic optimal solution. Consequently, their analysis does not carry over straightforwardly to the unregularised setting. One of the main challenges in our proof is to construct a different stopping time, independent of optimal parameters, such that the stopping time still occurs with small probability given a large enough batch size. We again first discuss the simultaneous approach followed by the dynamic approach.

Simultaneous stochastic policy gradient estimator: Consider K trajectories $(s_h^i, a_h^i)_{h=0}^{H-1}$, for $i = 1, \dots, K$, generated by $s_0^i \sim \mu$, $a_h^i \sim \pi^\theta(\cdot|s_h^i)$ and $s_h^i \sim p(\cdot|s_{h-1}^i, a_{h-1}^i)$ for $0 < h < H$. The gradient estimator is defined by

$$\widehat{\nabla} J^K(\theta, \mu) = \frac{1}{K} \sum_{i=1}^{K} \sum_{h=0}^{H-1} \nabla \log(\pi^\theta(a_h^i|s_h^i)) \hat{R}_h^i, \tag{7}$$

where $\hat{R}_h^i = \sum_{k=h}^{H-1} r(s_k^i, a_k^i)$ is an unbiased estimator of the h-state-action value function in (s_h^i, a_h^i) under policy π^θ. This gradient estimator is unbiased and has bounded variance (Lemma D.1). Then the stochastic PG updates for training the softmax parameter are given by

$$\bar{\theta}^{(n+1)} = \bar{\theta}^{(n)} + \eta \widehat{\nabla} J^K(\bar{\theta}^{(n)}, \mu). \tag{8}$$

Our main result for the simultaneous stochastic PG scheme is given as follows.

Theorem 4.1. *Under Assumption 3.1, let μ be a probability measure such that $\mu(s) > 0$ for all $s \in \mathcal{S}$. Consider the final policy using Algorithm 1 with stochastic updates from Equation 8 denoted by $\hat{\pi}^* = \pi^{\bar{\theta}^{(N)}}$. Moreover, for any $\delta, \epsilon > 0$ assume that the number of training steps satisfies $N \geq \left(\frac{21|\mathcal{S}|H^5 R^*}{\epsilon \delta c^2}\right)^2 \left\| \frac{d_\mu^{\pi^*}}{\mu} \right\|_\infty^4$, let $\eta = \frac{1}{5H^2 R^* \sqrt{N}}$ and $K \geq \frac{10 \max\{R^*, 1\}^2 N^3}{c^2 \delta^2}$. Then it holds true that*

$$\mathbb{P}\big(V_0^*(\mu) - V_0^{\hat{\pi}^*}(\mu) < \epsilon\big) > 1 - \delta\,.$$

Dynamic stochastic policy gradient estimator: For fixed h consider K_h trajectories $(s_k^i, a_k^i)_{k=h}^{H-1}$, for $i = 1, \dots, K_h$, generated by $s_h^i \sim \mu_h$, $a_h^i \sim \pi^\theta$ and $a_k^i \sim \tilde{\pi}_k$ for $h < k < H$. The estimator is defined by

$$\widehat{\nabla} J_h^K(\theta, \tilde{\pi}_{(h+1)}, \mu_h) = \frac{1}{K_h} \sum_{i=1}^{K_h} \nabla \log(\pi^\theta(a_h^i | s_h^i)) \hat{R}_h^i, \tag{9}$$

where $\hat{R}_h^i = \sum_{k=h}^{H-1} r(s_k^i, a_k^i)$ is an unbiased estimator of the h-state-action value function in (s_h^i, a_h^i) under policy $\tilde{\pi}$. Then the stochastic PG updates for training the parameter θ_h are given by

$$\bar{\theta}_h^{(n+1)} = \bar{\theta}_h^{(n)} + \eta_h \widehat{\nabla} J_h^{K_h}(\bar{\theta}_h^{(n)}, \tilde{\pi}_{(h+1)}, \mu_h). \tag{10}$$

Our main result for the dynamic stochastic PG scheme is given as follows.

Theorem 4.2. *For all $h \in \mathcal{H}$, let μ_h be probability measures such that $\mu_h(s) > 0$ for all $h \in \mathcal{H}$, $s \in \mathcal{S}_h$. Consider the final policy using Algorithm 2 with stochastic updates from Equation 10 denoted by $\hat{\pi}^* = (\pi^{\bar{\theta}_0^{(N_0)}}, \dots, \pi^{\bar{\theta}_{H-1}^{(N_{H-1})}})$. Moreover, for any $\delta, \epsilon > 0$ assume that the numbers of training steps satisfy $N_h \geq \left(\frac{12(H-h)R^* H^2 \left\| \frac{1}{\mu_h} \right\|_\infty}{\delta c_h^2 \epsilon} \right)^2$, let $\eta_h = \frac{1}{2(H-h)R^* \sqrt{N_h}}$ and $K_h \geq \frac{5 N_h^3 H^2}{c_h^2 \delta^2}$. Then it holds true that*

$$\mathbb{P}\Big(\forall s \in \mathcal{S}_0 : V_0^*(s) - V_0^{\hat{\pi}^*}(s) < \epsilon \Big) > 1 - \delta.$$

Comparison In both scenarios the derived complexity bounds for the stochastic PG uses a very large batch size and small step size. It should be noted that the choice of step size and batch size are closely connected and both strongly depend on the number of training steps N. Specifically, as N increases, the batch size increases, while the step size tends to decrease to prevent exceeding the stopping time with high probability. However, it is possible to increase the batch size even further and simultaneously benefit from choosing a larger step size, or vice versa.

An advantage of the dynamic approach is that c_h can be explicitly known for uniform initialisation. Hence, the complexity bounds for the dynamic approach results in a practicable algorithm, while c is unknown and possibly arbitrarily small for the simultaneous approach. Finally, we will also compare the complexity with respect to the time horizon. For the simultaneous approach the number of training steps scales with H^{10}, and the batch size with H^{30}, while in the dynamic approach the overall number of training steps scale with H^7 and the batch size with H^{20}. We are aware that these bounds are far from tight and irrelevant for practical implementations. Nevertheless, these bounds highlight once more the advantage of the dynamic approach in comparison to the simultaneous approach and show (the non-trivial fact) that the algorithms can be made to converge without knowledge of exact gradients and without regularisation.

5 Conclusion and Future Work

In this paper, we have presented a convergence analysis of two PG methods for undiscounted MDPs with finite-time horizon in the tabular parametrisation. Assuming exact gradients we have obtained an $\mathcal{O}(1/n)$-convergence rate for both approaches where the behavior regarding the time horizon and the model-dependent constant c is better in the dynamic approach than in the simultaneous approach. In the model-free setting we have derived complexity bounds to approximate the error to global optima with high probability using stochastic PG. It would be desirable to derive tighter bounds using for example adaptive step sizes or variance reduction methods that lead to more realistic batch sizes.

Similar to many recent results, the presented analysis relies on the tabular parametrisation. However, the heuristic from the policy gradient theorem does not, and the dynamic programming perspective suggests that parameters should be trained backwards in time. It would be interesting future work to see how this theoretical insight can be implemented in lower dimensional parametrisations using for instance neural networks.

ACKNOWLEDGMENTS

Special thanks to the anonymous reviewers for their constructive feedback and insightful discussions, which greatly improved this paper. We also acknowledge the valuable input received from the anonymous reviewers of previous submissions. Sara Klein thankfully acknowledges the funding support by the Hanns-Seidel-Stiftung e.V. and is grateful to the DFG RTG1953 "Statistical Modeling of Complex Systems and Processes" for funding this research.

REFERENCES

Tarek Abu Zwaida, Chuan Pham, and Yvan Beauregard. Optimization of inventory management to prevent drug shortages in the hospital supply chain. *Applied Sciences*, 11(6), 2021. URL https://www.mdpi.com/2076-3417/11/6/2726.

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. URL http://jmlr.org/papers/v22/19-736.html.

Carlo Alfano and Patrick Rebeschini. Linear convergence for natural policy gradient with log-linear policy parametrization. *arXiv Preprint*, arXiv:2209.15382, 2023. URL https://arxiv.org/abs/2209.15382.

Carlo Alfano, Rui Yuan, and Patrick Rebeschini. A novel framework for policy mirror descent with general parameterization and linear convergence. In *Advances in Neural Information Processing Systems*, volume 36, pp. 30681–30725. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/61a9278dfef5f871b5e472389f8d6fa1-Paper-Conference.pdf.

J. Bagnell, Sham M Kakade, Jeff Schneider, and Andrew Ng. Policy search by dynamic programming. In S. Thrun, L. Saul, and B. Schölkopf (eds.), *Advances in Neural Information Processing Systems*, volume 16. MIT Press, 2003. URL https://proceedings.neurips.cc/paper_files/paper/2003/file/3837a451cd0abc5ce4069304c5442c87-Paper.pdf.

Amir Beck. *First-Order Methods in Optimization*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2017. URL https://doi.org/10.1137/1.9781611974997.

Sebastian Becker, Patrick Cheridito, and Arnulf Jentzen. Deep optimal stopping. *Journal of Machine Learning Research*, 20(74):1–25, 2019. URL http://jmlr.org/papers/v20/18-232.html.

Dimitri P. Bertsekas and John N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1st edition, 1996. ISBN 1886529108.

Jalaj Bhandari and Daniel Russo. On the linear convergence of policy gradient methods for finite MDPs. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 2386–2394. PMLR, 13–15 Apr 2021. URL https://proceedings.mlr.press/v130/bhandari21a.html.

Jalaj Bhandari and Daniel Russo. Global optimality guarantees for policy gradient methods. *arXiv Preprint*, arXiv:1906.01786, 2022. URL https://arxiv.org/abs/1906.01786.

Shicong Cen, Chen Cheng, Yuxin Chen, Yuting Wei, and Yuejie Chi. Fast global convergence of natural policy gradient methods with entropy regularization. *Operations Research*, 70(4):2563–2578, 2022. URL https://doi.org/10.1287/opre.2021.2151.

Ignasi Clavera, Jonas Rothfuss, John Schulman, Yasuhiro Fujita, Tamim Asfour, and Pieter Abbeel. Model-based reinforcement learning via meta-policy optimization. In *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pp. 617–629. PMLR, 29–31 Oct 2018. URL https://proceedings.mlr.press/v87/clavera18a.html.

Yuhao Ding, Junzi Zhang, and Javad Lavaei. Beyond exact gradients: Convergence of stochastic soft-max policy gradient methods with entropy regularization. *arXiv Preprint*, arXiv:2110.10117, 2022. URL https://arxiv.org/abs/2110.10117.

Ilyas Fatkhullin, Anas Barakat, Anastasia Kireeva, and Niao He. Stochastic policy gradient methods: Improved sample complexity for Fisher-non-degenerate policies. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 9827–9869. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/fatkhullin23a.html.

Maryam Fazel, Rong Ge, Sham Kakade, and Mehran Mesbahi. Global convergence of policy gradient methods for the linear quadratic regulator. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1467–1476. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/fazel18a.html.

Soumyajit Guin and Shalabh Bhatnagar. A policy gradient approach for finite horizon constrained Markov decision processes. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pp. 3353–3359, 2023. URL https://ieeexplore.ieee.org/abstract/document/10383413.

Xin Guo, Anran Hu, and Junzi Zhang. Theoretical guarantees of fictitious discount algorithms for episodic reinforcement learning and global convergence of policy gradient methods. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(6):6774–6782, Jun. 2022. URL https://ojs.aaai.org/index.php/AAAI/article/view/20633.

Ben Hambly, Renyuan Xu, and Huining Yang. Policy gradient methods for the noisy linear quadratic regulator over a finite horizon. *SIAM Journal on Control and Optimization*, 59(5):3359–3391, 2021. URL https://doi.org/10.1137/20M1382386.

Ben Hambly, Renyuan Xu, and Huining Yang. Policy gradient methods find the Nash equilibrium in n-player general-sum linear-quadratic games. *Journal of Machine Learning Research*, 24(139): 1–56, 2023. URL http://jmlr.org/papers/v24/21-0842.html.

Feihu Huang, Shangqian Gao, Jian Pei, and Heng Huang. Momentum-based policy gradient methods. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4422–4433. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/huang20a.html.

Feihu Huang, Shangqian Gao, and Heng Huang. Bregman gradient policy optimization. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=ZU-zFnTum1N.

Emmeran Johnson, Ciara Pike-Burke, and Patrick Rebeschini. Optimal convergence rate for exact policy mirror descent in discounted Markov decision processes. In *Advances in Neural Information Processing Systems*, volume 36, pp. 76496–76524. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/f0d7b528c31bc3f9a0d5bab515ed6ed5-Paper-Conference.pdf.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pp. 267–274. Morgan Kaufmann Publishers Inc., 2002.

Sham M Kakade. A natural policy gradient. In *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001. URL https://proceedings.neurips.cc/paper_files/paper/2001/file/4b86abe48d358ecf194c56c69108433e-Paper.pdf.

Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the Polyak-Łojasiewicz condition. In *Machine Learning and Knowledge Discovery in Databases*, pp. 795–811, Cham, 2016. Springer International Publishing. URL https://doi.org/10.1007/978-3-319-46128-1_50.

Vijay Konda and John Tsitsiklis. Actor-critic algorithms. In *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999. URL https://proceedings.neurips.cc/paper_files/paper/1999/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

Jincheng Mei, Chenjun Xiao, Csaba Szepesvari, and Dale Schuurmans. On the global convergence rates of softmax policy gradient methods. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 6820–6829. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/mei20b.html.

Jincheng Mei, Yue Gao, Bo Dai, Csaba Szepesvari, and Dale Schuurmans. Leveraging non-uniformity in first-order non-convex optimization. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 7555–7564. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/mei21a.html.

Yurii Nesterov. *Introductory Lectures on Convex Optimization*. Springer New York, NY, 2013. URL https://doi.org/10.1007/978-1-4419-8853-9.

Matteo Papini, Damiano Binaghi, Giuseppe Canonaco, Matteo Pirotta, and Marcello Restelli. Stochastic variance-reduced policy gradient. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4026–4035. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/papini18a.html.

Matteo Pirotta, Marcello Restelli, and Luca Bascetta. Adaptive step-size for policy gradient methods. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper_files/paper/2013/file/f64eac11f2cd8f0efa196f8ad173178e-Paper.pdf.

B.T. Polyak. Gradient methods for the minimisation of functionals. *USSR Computational Mathematics and Mathematical Physics*, 3(4):864–878, 1963. URL https://doi.org/10.1016/0041-5553(63)90382-3.

M.L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2005. ISBN 9780470316887.

Bruno Scherrer. Approximate policy iteration schemes: A comparison. In Eric P. Xing and Tony Jebara (eds.), *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 1314–1322, Bejing, China, 22–24 Jun 2014. PMLR. URL https://proceedings.mlr.press/v32/scherrer14.html.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1889–1897, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/schulman15.html.

Zebang Shen, Alejandro Ribeiro, Hamed Hassani, Hui Qian, and Chao Mi. Hessian aided policy gradient. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5729–5738. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/shen19d.html.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999. URL https://proceedings.neurips.cc/paper_files/paper/1999/file/464d828b85b0bed98e80ade0a5c43b0f-Paper.pdf.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, 1992. URL https://doi.org/10.1007/BF00992696.

Lin Xiao. On the convergence rates of policy gradient methods. *Journal of Machine Learning Research*, 23(282):1–36, 2022. URL http://jmlr.org/papers/v23/22-0056.html.

Pan Xu, Felicia Gao, and Quanquan Gu. Sample efficient policy gradient methods with recursive variance reduction. In *International Conference on Learning Representations*, 2020a. URL https://openreview.net/forum?id=HJlxIJBFDr.

Pan Xu, Felicia Gao, and Quanquan Gu. An improved convergence analysis of stochastic variance-reduced policy gradient. In *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*, pp. 541–551. PMLR, 22–25 Jul 2020b. URL https://proceedings.mlr.press/v115/xu20a.html.

Rui Yuan, Robert M. Gower, and Alessandro Lazaric. A general sample complexity analysis of vanilla policy gradient. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera (eds.), *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pp. 3332–3380. PMLR, 28–30 Mar 2022. URL https://proceedings.mlr.press/v151/yuan22a.html.

Kaiqing Zhang, Xiangyuan Zhang, Bin Hu, and Tamer Basar. Derivative-free policy optimization for linear risk-sensitive and robust control design: Implicit regularization and sample complexity. In *Advances in Neural Information Processing Systems*, volume 34, pp. 2949–2964. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/1714726c817af50457d810aae9d27a2e-Paper.pdf.

Xiangyuan Zhang and Tamer Başar. Revisiting lqr control from the perspective of receding-horizon policy gradient. *IEEE Control Systems Letters*, 7:1664–1669, 2023. URL https://doi.org/10.1109/LCSYS.2023.3271594.

Xiangyuan Zhang, Bin Hu, and Tamer Başar. Learning the Kalman filter with fine-grained sample complexity. In *2023 American Control Conference (ACC)*, pp. 4549–4554, 2023a. URL https://doi.org/10.23919/ACC55779.2023.10156641.

Xiangyuan Zhang, Saviz Mowlavi, Mouhacine Benosman, and Tamer Başar. Global convergence of receding-horizon policy search in learning estimator designs. *arXiv Preprint*, arXiv:2309.04831, 2023b. URL https://arxiv.org/abs/2309.04831.