PILOT: AN $\mathcal{O}(1/K)$-CONVERGENT APPROACH FOR POLICY EVALUATION WITH NONLINEAR FUNCTION APPROXIMATION

Zhuqing Liu[†], **Xin Zhang**[‡], **Jia Liu**[†], **Zhengyuan Zhu**[‡], **Songtao Lu**[⋆]
[†]Department of Electrical and Computer Engineering, The Ohio State University
[‡]Department of Statistics, Iowa State University
[⋆]IBM Research, IBM Thomas J. Watson Research Center
liu.9384@osu.edu, xinzhang@iastate.edu, liu@ece.osu.edu,
zhuz@iastate.edu, songtao@ibm.com,

ABSTRACT

Learning an accurate value function for a given policy is a critical step in solving reinforcement learning (RL) problems. So far, however, the convergence speed and sample complexity performances of most existing policy evaluation algorithms remain unsatisfactory, particularly with *non-linear* function approximation. This challenge motivates us to develop a new path-integrated primal-dual stochastic gradient (PILOT) method, that is able to achieve a fast convergence speed for RL policy evaluation with nonlinear function approximation. To further alleviate the periodic full gradient evaluation requirement, we further propose an enhanced method with an adaptive-batch adjustment called PILOT$^+$. The main advantages of our methods include: i) PILOT allows the use of *constant* step sizes and achieves the $\mathcal{O}(1/K)$ convergence rate to first-order stationary points of non-convex policy evaluation problems; ii) PILOT is a generic *single*-timescale algorithm that is also applicable for solving a large class of non-convex strongly-concave minimax optimization problems; iii) By adaptively adjusting the batch size via historical stochastic gradient information, PILOT$^+$ is more sample-efficient empirically without loss of theoretical convergence rate. Our extensive numerical experiments verify our theoretical findings and showcase the high efficiency of the proposed PILOT and PILOT$^+$ algorithms compared with the state-of-the-art methods.

1 INTRODUCTION

In recent years, reinforcement learning (RL) has achieved enormous successes in a large number of areas, including healthcare (Petersen et al., 2019; Raghu et al., 2017b), financial recommendation (Theocharous et al., 2015), ranking system (Wen et al., 2023), resources management (Mao et al., 2016) and robotics (Levine et al., 2016; Raghu et al., 2017a), to name just a few. In RL, an agent interacts with an environment and repeats the tasks of observing the current state, performing a policy-based action, receiving a reward, and transitioning to the next state. Upon collecting a trajectory of action-reward sample pairs, the agent updates its policy with the aim of maximizing its long-term accumulative reward. In this RL framework, a key step is the policy evaluation (PE) problem, which aims to learn the value function that estimates the expected long-term accumulative reward for a given policy. Value functions not only explicitly provide the agent's accumulative rewards, but are also able to update the current policy so that the agent can visit valuable states more frequently (Lagoudakis & Parr, 2003). Regarding PE, two of the most important performance metrics are *convergence rate* and *sample complexity*. First, since PE is a subroutine of an overall RL task, developing fast-converging PE algorithms is of critical importance to the overall efficiency of RL. Second, due to the challenges in collecting a large number of training samples (trajectories of state-action pairs) for PEs in RL, reducing the number of samples (i.e., sample complexity) can significantly alleviate the burden of data collection for solving PE problems. These two important aspects motivate us to pursue a fast-converging PE algorithm with a low sample complexity in this work.

Among various algorithms for PE, one of the simplest and most effective methods is the temporal difference (TD) learning approach (Sutton, 1988). In TD learning, instead of focusing on the predicted and actual outcomes, the key idea is to make the difference between temporally successive predictions small. Specifically, the TD learning approach learns the value function using the Bellman equation to bootstrap from the currently estimated value function. To date, there have been many algorithms proposed within the family of TD learning (Dann et al., 2014). However, most of these methods suffer from either unstable convergence performance, (e.g., TD(λ) (Sutton, 1988) for off-policy training) or high computational complexity (e.g., the least-squares temporal difference (LSTD) (Boyan, 2002)) in training with massive features. One reason of the unstable convergence performance of these early attempts is that they do not leverage the gradient-oracle in PE. Thus, in recent years, gradient-based PE algorithms have attracted increasing attention.

However, when working with nonlinear DNN models, the convergence performance of the conventional *single-timescale* TD algorithms may not be guaranteed (Tsitsiklis & Van Roy, 1996). To address this issue, some convergent two-timescale algorithms (Maei et al., 2009; Chung et al., 2018) have been proposed at the expense of higher implementation complexity. Second, modern PE tasks could involve a large amount of state transition data. To perform PE, algorithms typically need to calculate *full gradients* that require all training data (e.g., gradient temporal difference (GTD) (Sutton et al., 2008) and TD with gradient correction (TDC) (Sutton et al., 2009)], which entails a high sample complexity. To the best of our knowledge, all existing works on PE either focus on linear approximation, such as GTD2 (Sutton et al., 2009), PDBG (Du et al., 2017), SVRG (Du et al., 2017), SAGA (Du et al., 2017) or they exhibit slower theoretical convergence performance, as observed in STSG (Qiu et al., 2020), VR-STSG (Qiu et al., 2020), nPD-VR (Wai et al., 2019) in the sense of achieving a convergence rate that is slower than $O(1/K)$, where K is the number of iterations. Please see detailed discussions in Section. 2. In light of the above limitations, in this paper, we ask the following critical question:

Can we develop a fast-converging single-timescale algorithm for PE with nonlinear function approximation?

In this paper, we give an *affirmative* answer to the above question. Specifically, we propose an efficient path-integrated primal-dual stochastic gradient algorithm (PILOT) to tackle the PE problem with nonlinear function approximation, which we recast as a minimax optimization problem. The proposed PILOT algorithm admits a simple and elegant *single-timescale* algorithmic structure. Besides, we further enhance PILOT by proposing PILOT$^+$, which uses adaptive batch sizes to avoid the periodic full gradient evaluation to further reduce sample complexity. The major contribution of this paper is that our proposed algorithms achieve the first $\mathcal{O}(1/K)$ convergence rate (K is the number of iterations) with *constant step-sizes* for PE with *nonlinear* function approximation, which is the *best* result in the literature so far. Our main results are highlighted below:

- Utilizing a variance reduction technique, our PILOT algorithm facilitates the use of constant step-sizes while maintaining a low sample complexity. We demonstrate that, under reasonable mild assumptions and suitable parameter selections, PILOT attains an $\mathcal{O}(1/K)$ convergence rate to a first-order stationary point for a class of nonconvex-strongly-concave (NCX-SCV) minimax problems encountered in RL. To establish this outcome, our convergence analysis employs new proof techniques and resolves an ambiguity present in the current state-of-the-art convergence analyses of Variance Reduction (VR)-based PE methods.

- Our PILOT$^+$ algorithm leverages adaptive batch sizes, effectively integrating historical information throughout the optimization process without necessitating backtracking or condition verification. We demonstrate that PILOT$^+$ leads to a substantial reduction in both sample requirements and gradient computation loads. This reduction is made possible by our innovative adaptive batch size technique, which eliminates the need for full gradient evaluation.

- Our comprehensive experimental results provide strong evidence of the superior performance of our algorithms compared to state-of-the-art gradient-based PE methods. Additionally, PILOT$^+$ exhibits the capability to further reduce the sample complexity of the PILOT algorithm. It is worth noting that while our primary focus is on PE, the design of our algorithms and the proof techniques developed also hold potential significance in the broader domain of minimax optimization, presenting independent theoretical interest.

Table 1: PE algorithms comparison: M is the size of the dataset and K is the total iteration.

Algorithm	Function Approx.	Problem	Step-size	Convergence Rate
GTD2 Sutton et al. (2009)	Linear	-	$\mathcal{O}(1)$	-
PDBG Du et al. (2017)	Linear	Convex-Concave	$\mathcal{O}(1)$	$\mathcal{O}(1/K)$
SVRG Du et al. (2017)	Linear	Convex-Concave	$\mathcal{O}(1)$	$\mathcal{O}(1/K)$
SAGA Du et al. (2017)	Linear	Convex-Concave	$\mathcal{O}(1)$	$\mathcal{O}(1/K)$
TATD Patil et al. (2023)	Linear	-	-	$\mathcal{O}(1/K)$
STSG Qiu et al. (2020)	Nonlinear	Stochastic/ NCX-SCV	$\mathcal{O}(1)$	$\mathcal{O}(1/K^{1/2})$
VR-STSG Qiu et al. (2020)	Nonlinear	Stochastic/ NCX-SCV	$\mathcal{O}(1)$	$\mathcal{O}(1/K^{2/3})$
nPD-VR Wai et al. (2019)	Nonlinear	Finte-Sum / NCX-SCV	$\mathcal{O}(1/M)$	Slower than[1] $\mathcal{O}(1/K)$
PILOT [Ours.]	Nonlinear	Finte-Sum / NCX-SCV	$\mathcal{O}(1)$	$\mathcal{O}(1/K)$
PILOT$^+$ [Ours.]	Nonlinear	Finte-Sum / NCX-SCV	$\mathcal{O}(1)$	$\mathcal{O}(1/K)$

[1] The convergence rate of nPD-VR is ambiguous. See the detailed discussions in Sections 2 below and 4.

2 Related work

1) TD Learning with Function Approximation for PE: TD learning with function approximation plays a vital role in PE. The key idea of TD learning is to minimize the Bellman error for approximating the value function. However, most existing TD learning algorithms with theoretical guarantees focus on *the linear approximation setting* (e.g., (Sutton et al., 2008; Srikant & Ying, 2019; Xu et al., 2019; Touati et al., 2018; Patil et al., 2023; Li et al., 2021)). Existing works in (Doan et al., 2019; Liu et al., 2015; Macua et al., 2014; Zhang & Xiao, 2019; Patil et al., 2023; Li et al., 2021) provided a finite-time analysis for the distributed TD(0) and showed that the convergence rates of their algorithms are $\mathcal{O}(1/K)$. It was shown in (Du et al., 2017) that PE with linear function approximation by TD(0) can be formulated as a strongly convex-concave or convex-concave problem, and can be solved by a primal-dual method with a linear convergence rate. Unfortunately, the linearity assumption cannot be applied to a wide range of PE problems with nonlinear models. TD learning with nonlinear (smooth) function approximation is far more complex. The work in (Maei et al., 2009) was among the first to propose a general framework for minimizing the generalized mean-squared projected Bellman error (MSPBE) with smooth and *nonlinear* value functions. Despite their use of *two-timescale* step-sizes, it's important to note that this approach yielded slow convergence performance. Other TD methods with nonlinear function approximations for PE include (Wang et al., 2017; 2016). Nonlinear TD learning was also investigated in Qiu et al. (2020), which proposed two single-timescale first-order stochastic algorithms. However, the convergence rates of their STSG and VR-STSG methods are $\mathcal{O}(1/K^{1/4})$ and $\mathcal{O}(1/K^{1/3})$, while our PILOT algorithm achieves a *much faster* $\mathcal{O}(1/K)$ convergence rate, matching the standard one in the linear case.

In PE with non-linear function approximation, the state-of-the-art and the most related work is (Wai et al., 2019), which showed that minimizing the generalized MSPBE problem is equivalent to solving a non-convex-strongly-concave (NCX-SCV) minimax optimization problem by applying the Fenchel's duality. However, their best convergence results only hold for a small step-size that is $\mathcal{O}(1/M)$, where M denotes the size of the dataset. This will be problematic for RL problems with a large state-action transition dataset. Furthermore, it is worth highlighting that their convergence rate bound takes the form of $\frac{F^{(K)}+\text{Constant1}}{K\cdot\text{Constant2}}$ (cf. Theorem 1, Eq. (26) in Wai et al. (2019)). Here, the term $F^{(K)}$ in the denominator inherently relies on the primal and dual values $\boldsymbol{\theta}^{(K)}$ and $\boldsymbol{\omega}^{(K)}$ at the K-th iteration. However, the bounding of $\boldsymbol{\omega}^{(K)}$ in (Wai et al., 2019) remains unclear, leading to ambiguity when attempting to guarantee an $\mathcal{O}(1/K)$ convergence rate. Therefore, whether an $\mathcal{O}(1/K)$ convergence rate is achievable in single-timescale PE with nonlinear function approximation and constant step-sizes remains *an open question* thus far. **The key contribution and novelty** in this paper is that we resolve the above open question by proposing two new algorithms, both achieving an $\mathcal{O}(1/K)$ convergence rate. To establish this result, we propose a **new convergence metric** (cf. Eq. (6) in Section 4), which necessitates new proof techniques and analysis. For easy comparisons, we summarize our algorithms in a comparison with the related existing works in Table 1.

2) Relations with NCX-SCV minimax Optimization: Although the focus of our paper is on PE, our algorithmic techniques are also related to the general area of NCX-SCV minimax optimization due to the primal-dual MSPBE formulation (cf. Eq. (1) in Section 3) [1]. Early attempts in (Nouiehed

[1]We note that solving the MSPBE-based PE problem is one of the most fitting motivating applications for NCX-SCV minimax optimization problems. As discussed in Section 3, the MSPBE-based PE problem

et al., 2019; Lin et al., 2020b; Lu et al., 2019) developed gradient descent-ascent algorithms to solve the NCX-SCV minimax problems. However, these methods suffer from a high sample complexity and slow convergence rate. To overcome this limitation, two variance-reduction algorithms named SREDA (Luo et al., 2020) are proposed for solving NCX-SCV minimax problems, which share some similarities to our work. SREDA was later enhanced in Xu et al. (2020) to allow larger step-sizes. However, our algorithms differ from SREDA in the following key aspects: (i) Our algorithms are *single-timescale* algorithms (see Section 4 for the notions of single-timescale and two-timescale algorithms), which are much easier to implement. In comparison, SREDA is a two-timescale algorithm that also requires solving an inner concave maximization subproblem. To a certain extent, SREDA can be viewed as a triple-loop structure, and hence the implementation complexity of SREDA is higher than ours; (ii) In the initialization stage, SREDA uses a subroutine called PiSARAH to help the SREDA algorithm achieve the desired accuracy at the initialization step and can be seen as an additional step to solve an inner concave maximization subproblem. Thus, SREDA has a higher computation cost than our algorithm. (iii) The number of hyperparameters in SREDA is more than ours and it requires the knowledge of the condition number to set the algorithm's parameters for good convergence performance. By contrast, our algorithms only require step-sizes α and β to be sufficiently small (smaller than the upper bounds we provide in our theorems), which is easier to tune in practice. (iv) SREDA does *not* provide an explicit convergence rate in their paper (it is also unclear what their convergence rate is from their proof). Yet, we show that our PILOT algorithm has a lower sample complexity than that of SREDA.

Another related work in NCX-SCV minimax optimization is (Zhang et al., 2021), which provided sample complexity upper and lower bounds. However, there remains a gap between the sample complexity lower and upper bounds in (Zhang et al., 2021). By contrast, the sample complexity of our PILOT algorithm *is the first to match the lower bound* $\mathcal{O}(M + \sqrt{M}\epsilon^{-2})$ in (Zhang et al., 2021)[2]. Furthermore, the algorithm in (Zhang et al., 2021) contains an inner minimax subproblem (cf. Line 6 of Algorithm 1 in Zhang et al. (2021)). Solving such a subproblem in the inner loop incurs high computational costs. Due to this reason, the algorithm in (Zhang et al., 2021) had to settle for an inexact solution, which hurts the convergence performance in practice. In contrast, our algorithm does not have such a limitation.

3 PRELIMINARIES AND PROBLEM STATEMENT

We start by introducing the necessary background of RL, with a focus on the PE problem based on nonlinear function approximation.

1) Policy Evaluation with Nonlinear Function Approximation: RL problems are formulated using the Markov decision process (MDP) framework defined by a five-tuple $\{\mathcal{S}, \mathcal{A}, P, \gamma, \mathcal{R}\}$, where $\mathcal{S}$ denotes the state space and $\mathcal{A}$ is the action space; $P : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ represents state transition probabilities after taking actions; $\mathcal{R}$ denotes the space of the received rewards, where each reward corresponds to the agent taking a specific action $a \in \mathcal{A}$ from the set of possible actions when the system is in a particular state in the state space $s \in \mathcal{S}$ (in this paper, we assume that the state and action spaces are finite but their dimensionality could be large); and $\gamma \in [0, 1)$ is a time-discount factor. For RL problems over an infinite discrete-time horizon $\{t \in \mathbb{N}\}$, the learning agent executes an action a_t according to the state s_t and some policy $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$. The system then transitions into a new state s_{t+1} in the next time slot, and the agent receives a reward $R(s_t, a_t)$ through its interaction with the environment. The trajectory generated by a policy π is a sequence of state-action pairs denoted as $\{s_1, a_1, s_2, a_2, \ldots\}$. Specifically, for a policy π (could be a randomized policy), the expected reward received by the agent at state s in any given time slot can be computed as $R^\pi(s_t) = \mathbb{E}_{a\sim\pi(\cdot|s)}\big[R^\pi(s_t, a)\big]$. The value function $V^\pi(s_0) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t) \mid s_0, \pi\right]$ indicates the long-term discounted reward of policy π over an infinite horizon with the initial state at $s_0 \in \mathcal{S}$. Also, the Bellman equation implies that $V(\cdot)$ satisfies : $V(s) = \mathcal{T}^\pi V(s)$, where $\mathcal{T}^\pi f(s) \triangleq \mathbb{E}[R^\pi(s) + \gamma f(s')|a \sim \pi(\cdot|s), s' \sim P(\cdot|s, a)]$ denotes the Bellman operator. In RL, the agent's goal is to determine an optimal policy π^* that maximizes the value function $V^\pi(s)$ from any initial state s.

can be reformulated as an NCX-SCV minimax problem, which can be solved by our proposed VR-based single-timescale method efficiently.

[2]Here, the rate of $\mathcal{O}(\epsilon^{-2})$ measured on the size of the primal objective function is equivalent to $\mathcal{O}(1/K)$.

However, the first obstacle in solving RL problems stems from evaluating $V^\pi(\cdot)$ for a given π since $P(\cdot|s,a)$ is unknown. Moreover, it is often infeasible to store $V^\pi(s)$ since the state space $\mathcal{S}$ could be extremely large. To address these challenges, one popular approach in RL is to approximate $V^\pi(\cdot)$ using a family of parametric and smooth functions in the form of $V^\pi(\cdot) \approx V_{\boldsymbol{\theta}^\pi}(\cdot)$, where $\boldsymbol{\theta}^\pi \in \Theta \subseteq \mathbb{R}^d$ is a d-dimensional parameter vector. Here, Θ is a compact subspace. For notational simplicity, we will omit all superscripts "π" whenever the policy π is clear from the context. In this paper, we focus on **nonlinear function approximation**, i.e., $V_{\boldsymbol{\theta}}(\cdot) : \mathcal{S} \to \mathbb{R}$ is a nonlinear function with respect to (w.r.t.) $\boldsymbol{\theta}$. For example, $V_{\boldsymbol{\theta}}(\cdot)$ could be based on a $\boldsymbol{\theta}$-parameterized nonlinear DNN. We assume that the gradient and Hessian of $V_{\boldsymbol{\theta}}(\cdot)$ exist and are denoted as: $g_{\boldsymbol{\theta}}(s) := \nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}}(s) \in \mathbb{R}^d$, $H_{\boldsymbol{\theta}}(s) := \nabla^2_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}}(s) \in \mathbb{R}^{d\times d}$. Our goal is to find the optimal parameter $\boldsymbol{\theta}^* \in \mathbb{R}^d$ that minimizes the error between $V_{\boldsymbol{\theta}^*}(\cdot)$ and $V(\cdot)$. It has been shown in (Liu et al., 2018) that this problem can be formulated as minimizing the mean-squared projected Bellman error (MSPBE) of the value function (Liu et al., 2018, Proposition 1) as follows:

$$\begin{aligned}\text{MSPBE}(\boldsymbol{\theta}) &:= \frac{1}{2}\big\|\mathbb{E}_{\mathbf{s}\sim D^\pi(\cdot)}\big[\big(\mathcal{T}^\pi V_{\boldsymbol{\theta}}(s) - V_{\boldsymbol{\theta}}(s)\big)\nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}}(s)^\top\big]\big\|^2_{\boldsymbol{D}^{-1}} \\ &= \max_{\boldsymbol{\omega}\in\mathbb{R}^d}\Big(-\frac{1}{2}\mathbb{E}_{\mathbf{s}\sim D^\pi(\cdot)}[(\boldsymbol{\omega}^\top g_\theta(s))^2] + \langle \boldsymbol{\omega}, \mathbb{E}_{\mathbf{s}\sim D^\pi(\cdot)}\big[\big(\mathcal{T}^\pi V_{\boldsymbol{\theta}}(s) - V_{\boldsymbol{\theta}}(s)\big) g_{\boldsymbol{\theta}}(s)\big]\rangle\Big), \end{aligned} \tag{1}$$

where $D^\pi(\cdot)$ is the stationary distribution under policy π and $\boldsymbol{D} \triangleq \mathbb{E}_{\mathbf{s}\sim D^\pi}[g_{\boldsymbol{\theta}}(s) g^\top_{\boldsymbol{\theta}}(s)] \in \mathbb{R}^{d\times d}$ and $\boldsymbol{\omega}$ is referred to as the dual variable.

2) Primal-Dual Optimization for MSPBE: Minimizing MSPBE($\boldsymbol{\theta}$) in (1) is equivalent to solving a primal-dual minimax optimization problem: $\min_{\boldsymbol{\theta}\in\mathbb{R}^d}\max_{\boldsymbol{\omega}\in\mathbb{R}^d} L(\boldsymbol{\theta},\boldsymbol{\omega})$, where $L(\boldsymbol{\theta},\boldsymbol{\omega}) \triangleq \langle \boldsymbol{\omega}, \mathbb{E}_{\mathbf{s}\sim D^\pi(\cdot)}\big[\big(\mathcal{T}^\pi V_{\boldsymbol{\theta}}(s) - V_{\boldsymbol{\theta}}(s)\big) g_{\boldsymbol{\theta}}(s)^\top\big]\rangle - \frac{1}{2}\mathbb{E}_{\mathbf{s}\sim D^\pi(\cdot)}[(\boldsymbol{\omega}^\top g_{\boldsymbol{\theta}}(s))^2]$. Since the distribution $D^\pi(\cdot)$ is unknown and the expectation cannot be evaluated directly, most existing work in the literature (see, e.g., Liu et al. (2015); Wai et al. (2019); Du et al. (2017)) considered the following empirical minimax problem by replacing the expectation in $L(\boldsymbol{\theta},\boldsymbol{\omega})$ by a finite-sum sample average approximation [3] based on an M-step trajectory $\{s_1, a_1 \cdots, s_M, a_M, s_{M+1}\}$ generated by some policy π, i.e.,

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d}\max_{\boldsymbol{\omega}\in\mathbb{R}^d}\frac{1}{M}\sum_{i=1}^{M}\mathcal{L}_i(\boldsymbol{\theta},\boldsymbol{\omega}) = \min_{\boldsymbol{\theta}\in\mathbb{R}^d}\max_{\boldsymbol{\omega}\in\mathbb{R}^d}\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\omega}), \tag{2}$$

where $\mathcal{L}_i(\boldsymbol{\theta},\boldsymbol{\omega}) := \langle \boldsymbol{\omega}, [R(s_i,a_i,s_{i+1}) + \gamma V_{\boldsymbol{\theta}}(s_{i+1}) - V_{\boldsymbol{\theta}}(s_i)] \times g_{\boldsymbol{\theta}}(s_i)\rangle - \frac{1}{2}(\boldsymbol{\omega}^\top g_{\boldsymbol{\theta}}(s_i))^2$. In Appendix F, we will also discuss the minimax problem with $\theta \in \Theta, \omega \in \mathcal{W}$, where $\Theta, \mathcal{W}$ are convex constrained sets. Solving Problem (2) for MSPBE constitutes the rest of this paper.

Note that Problem (2) is non-convex in general (e.g., DNN-based nonlinear approximation). Let $J(\boldsymbol{\theta}) \triangleq \max_{\boldsymbol{\omega}\in\mathbb{R}^d}\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\omega})$. Then, we can equivalently rewrite Problem (2) as follows: $\min_{\boldsymbol{\theta}\in\mathbb{R}^d}\max_{\boldsymbol{\omega}\in\mathbb{R}^d}\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\omega}) = \min_{\boldsymbol{\theta}\in\mathbb{R}^d} J(\boldsymbol{\theta})$. Note from (2) that $\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\omega})$ is strongly concave w.r.t. $\boldsymbol{\omega}$, which guarantees the existence and uniqueness of the solution to the problem $\max_{\boldsymbol{\omega}\in\mathbb{R}^d}\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\omega}), \forall \boldsymbol{\theta} \in \mathbb{R}^d$. Then, given $\boldsymbol{\theta} \in \mathbb{R}^d$, we define the following notation: $\boldsymbol{\omega}^*(\boldsymbol{\theta}) := \text{argmax}_{\boldsymbol{\omega}\in\mathbb{R}^d}\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\omega})$. Subsequently, $J(\boldsymbol{\theta})$ can be further written as $J(\boldsymbol{\theta}) = \mathcal{L}(\boldsymbol{\theta},\boldsymbol{\omega}^*(\boldsymbol{\theta}))$. We aim to minimize $J(\boldsymbol{\theta})$ by finding the stationary point of $\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\omega})$. For the sake of simplicity in notation, we use $\boldsymbol{\omega}^*$ to denote $\boldsymbol{\omega}^*(\boldsymbol{\theta})$. Note that if $\boldsymbol{D}$ in Eq. (1) is positive definite, Problem (2) is strongly concave in $\boldsymbol{\omega}$, but non-convex in $\boldsymbol{\theta}$ in general due to the non-convexity of function $V_{\boldsymbol{\theta}}$. Thus, the Problem 2 is an NCX-SCV optimization problem.

3) Sample Complexity: In this paper, we adopt the following sample complexity metric to measure the dat efficiency of an optimization algorithm, which is widely used in the literature (e.g., Luo et al. (2020); Zhang et al. (2021); Xu et al. (2020)):

Definition 1 (Sample Complexity)**.** The sample complexity is defined as the total number of required samplings from the dataset to evaluate incremental first-order oracle (IFO) until an algorithm converges, where one IFO call evaluates a pair of $(\mathcal{L}_i(\theta,\omega), \nabla\mathcal{L}_i(\theta,\omega)), i \in [M]$.

[3] Although the finite-sum empirical loss is an approximation of the expected loss function for PE, as shown in Chen et al. (2021), under the conditions of bounded instantaneous loss and bounded derivatives (satisfied for most applications in practice), the approximation error of using empirical loss is small with high probability (cf. (Chen et al., 2021, Lemma 2)). Thus, the empirical loss has been widely used as a proxy for the expected loss in the literature Liu et al. (2015); Wai et al. (2019); Du et al. (2017).

Algorithm 1 The path-integrated primal-dual stochastic gradient (PILOT).

Input: An M-step trajectory of the state-action pairs $\{s_1, a_1, s_2, a_2, \cdots, s_M, a_M, s_{M+1}\}$ generated from a given policy; step sizes $\alpha, \beta \geq 0$; initialization points $\boldsymbol{\theta}^0 \in \mathbb{R}^d$, $\boldsymbol{\omega}^0 \in \mathbb{R}^d$.

Output: $(\boldsymbol{\theta}^{(\widetilde{K})}, \boldsymbol{\omega}^{(\widetilde{K})})$, where $\widetilde{K}$ is independently and uniformly picked from $\{1, \cdots, K\}$;

1: **for** $k = 0, 1, 2, \cdots, K-1$ **do**
2: If $\text{mod}(k, q) = 0$, compute full gradients $G_{\boldsymbol{\theta}}^{(k)}, G_{\boldsymbol{\omega}}^{(k)}$ as in Eq. (3). Otherwise, select $|\mathcal{N}|$
3: samples independently and uniformly from $[M]$, and compute gradients as in Eq. (4).
4: Perform the primal-dual updates to obtain the next iterate $\boldsymbol{\theta}^{(k+1)}, \boldsymbol{\omega}^{(k+1)}$ as in Eq. (5).
5: **end for**

4 The Variance-Reduced Primal-Dual Method (PILOT)

In this section, we first present the variance-reduced primal-dual (PILOT) algorithm for PE, followed by the theoretical convergence results. Due to space limitation, we provide a proof sketch in the main text and relegate the detailed proofs to the supplementary material.

1) Algorithm Description: The full description of PILOT is illustrated in Algorithm 1. In PILOT, for every q iterations, the algorithm calculates the full gradients as follows:

$$G_{\boldsymbol{\theta}}^{(k)} = \frac{1}{|M|} \sum_{i \in M} \nabla_{\boldsymbol{\theta}} \mathcal{L}_i(\boldsymbol{\theta}^{(k)}, \boldsymbol{\omega}^{(k)}), \quad G_{\boldsymbol{\omega}}^{(k)} = \frac{1}{|M|} \sum_{i \in M} \nabla_{\boldsymbol{\omega}} \mathcal{L}_i(\boldsymbol{\theta}^{(k)}, \boldsymbol{\omega}^{(k)}), \text{if } \text{mod}(k, q) = 0. \tag{3}$$

In all other iterations, PILOT selects a batch $\mathcal{N}$ and computes variance-reduced gradient estimators:

$$G_{\boldsymbol{\theta}}^{(k)} = \frac{1}{|\mathcal{N}|} \sum_{i \in \mathcal{N}} \big(\nabla_{\boldsymbol{\theta}} \mathcal{L}_i(\boldsymbol{\theta}^{(k)}, \boldsymbol{\omega}^{(k)}) - \nabla_{\boldsymbol{\theta}} \mathcal{L}_i(\boldsymbol{\theta}^{(k-1)}, \boldsymbol{\omega}^{(k-1)}) + G_{\boldsymbol{\theta}}^{(k-1)}\big), \text{if } \text{mod}(k, q) \neq 0, \tag{4a}$$

$$G_{\boldsymbol{\omega}}^{(k)} = \frac{1}{|\mathcal{N}|} \sum_{i \in \mathcal{N}} \big(\nabla_{\boldsymbol{\omega}} \mathcal{L}_i(\boldsymbol{\theta}^{(k)}, \boldsymbol{\omega}^{(k)}) - \nabla_{\boldsymbol{\omega}} \mathcal{L}_i(\boldsymbol{\theta}^{(k-1)}, \boldsymbol{\omega}^{(k-1)}) + G_{\boldsymbol{\omega}}^{(k-1)}\big), \text{if } \text{mod}(k, q) \neq 0. \tag{4b}$$

The estimators in (4) are constructed iteratively based on the previous update information $\nabla_{\boldsymbol{\theta}} \mathcal{L}_i(\boldsymbol{\theta}^{(k-1)}, \boldsymbol{\omega}^{(k-1)})$ (resp. $(\nabla_{\boldsymbol{\omega}} \mathcal{L}_i(\boldsymbol{\theta}^{(k-1)}, \boldsymbol{\omega}^{(k-1)})$) and $G_{\boldsymbol{\theta}}^{(k-1)}$ (resp. $G_{\boldsymbol{\omega}}^{(k-1)}$). PILOT updates the primal and dual variables as follows:

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - \beta G_{\boldsymbol{\theta}}^{(k)}, \quad \boldsymbol{\omega}^{(k+1)} = \boldsymbol{\omega}^{(k)} + \alpha G_{\boldsymbol{\omega}}^{(k)}, \tag{5}$$

where parameters α and β are the *constant* learning rates for the primal and dual updates.

Remark 1. Single-Timescale Algorithm: Our PILOT algorithm is a single-timescale algorithm, which is much simpler to implement in practice compared to the two-timescale algorithms shown in Maei et al. (2009); Lin et al. (2020b). To see this, we first restate the notions of single- and two-timescale algorithms in the literature (see, e.g., (Dalal et al., 2018)). Let $\alpha_t \geq 0$ and $\beta_t \geq 0$ represent the step-sizes at iteration t for outer- and inner-variable updates, respectively. An algorithm is called a two-timescale algorithm if $\alpha_t/\beta_t \to 0$ or $\alpha_t/\beta_t \to +\infty$ as $t \to \infty$. On the other hand, an algorithm is called a single-timescale algorithm if $0 < C \leq \alpha_t/\beta_t \leq C'$ as $t \to \infty$, where $0 < C, C' < +\infty$ are two positive constants. In our proposed PILOT algorithm, since the step-sizes α_t and β_t are constants, our PILOT algorithm is clearly a single-timescale algorithm.

2) Assumptions: Before showing the theoretical results, we first make the following assumptions:

Assumption 1 (μ-Strong Concavity). We assume that $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\omega})$ is differentiable and μ-strongly concave in $\boldsymbol{\omega}$, where for any $\boldsymbol{\theta} \in \mathbb{R}^d$, $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\omega}) \leq \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\omega}') + \nabla_{\boldsymbol{\omega}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\omega}')^\top (\boldsymbol{\omega} - \boldsymbol{\omega}') - \frac{\mu}{2} \|\boldsymbol{\omega} - \boldsymbol{\omega}'\|^2$, $\forall \boldsymbol{\omega}, \boldsymbol{\omega}' \in \mathbb{R}^d$, $\mu > 0$.

It can be shown that the condition in Assumption 1 is equivalent to : $\|\nabla_{\boldsymbol{\omega}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\omega}) - \nabla_{\boldsymbol{\omega}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\omega}')\| \geq \mu \|\boldsymbol{\omega} - \boldsymbol{\omega}'\|$, $\forall \boldsymbol{\omega}, \boldsymbol{\omega}' \in \mathbb{R}^d$ (see proofs in (Zhou, 2018, Lemmas 2 and 3)).

Assumption 2 (L_f-Smoothness). We assume that for $i = 1, 2, \ldots, M$, both gradient $\nabla_{\boldsymbol{\theta}} \mathcal{L}_i(\boldsymbol{\theta}, \boldsymbol{\omega})$ and $\nabla_{\boldsymbol{\omega}} \mathcal{L}_i(\boldsymbol{\theta}, \boldsymbol{\omega})$ are L_f-smooth. That is, for all $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathbb{R}^d$ and $\boldsymbol{\omega}, \boldsymbol{\omega}' \in \mathbb{R}^d$, there exists a constant $L_f > 0$ such that $\|\nabla \mathcal{L}_i(\boldsymbol{\theta}, \boldsymbol{\omega}) - \nabla \mathcal{L}_i(\boldsymbol{\theta}', \boldsymbol{\omega}')\| \leq L_f \left(\|\boldsymbol{\theta} - \boldsymbol{\theta}'\| + \|\boldsymbol{\omega} - \boldsymbol{\omega}'\|\right)$.

Assumption 3 (Boundness from Below). There exists a finite lower bound $J^* = \inf_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) > -\infty$.

Assumption 4 (Bounded Variance). There exists a constant $\sigma > 0$ such that for all $\boldsymbol{\theta} \in \mathbb{R}^d, \boldsymbol{\omega} \in \mathbb{R}^d$, $\frac{1}{M} \sum_{i=1}^M \|\nabla_{\boldsymbol{\theta}} \mathcal{L}_i(\boldsymbol{\theta}, \boldsymbol{\omega}) - \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\omega})\|^2 \leq \sigma^2$ and $\frac{1}{M} \sum_{i=1}^M \|\nabla_{\boldsymbol{\omega}} \mathcal{L}_i(\boldsymbol{\theta}, \boldsymbol{\omega}) - \nabla_{\boldsymbol{\omega}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\omega})\|^2 \leq \sigma^2$.

We note that Assumption 1 is satisfied if the number of samples M is sufficiently large and the matrix $\boldsymbol{D}$ is positive definite [4]. Assumption 3 is standard in the optimization literature. Assumption 4 is also commonly adopted for proving convergence results of SGD- and VR-based algorithms, or algorithms that draw a mini-batch of samples instead of all samples. Assumption 4 is guaranteed to hold under the compact set condition and common for stochastic approximation algorithms for minimax optimization (Qiu et al., 2020; Lin et al., 2020a). Assumptions 1–4 are also often used in temporal difference (TD) problems (see, e.g., (Qiu et al., 2020; Wai et al., 2019)). With these assumptions, we are now in a position to present our convergence performance results of PILOT.

3) Convergence Performance: In this paper, we propose a *new metric* for convergence analysis:

$$\mathfrak{M}(\boldsymbol{\theta}, \boldsymbol{\omega}) := \|\nabla J(\boldsymbol{\theta})\|^2 + 2\|\boldsymbol{\omega} - \boldsymbol{\omega}^*(\boldsymbol{\theta})\|^2. \tag{6}$$

The first term in (6) measures the convergence of the primal variable $\boldsymbol{\theta}$. As common in non-convex optimization analysis, $\|\nabla J(\boldsymbol{\theta})\|^2 = 0$ indicates that $\boldsymbol{\theta}$ is a first-order stationary point (FOSP) of Problem (2). The second term in (6) measures the convergence of $\boldsymbol{\omega}^{(k)}$ to the unique maximizer $\boldsymbol{\omega}^*(\boldsymbol{\theta}^{(k)})$ for $\mathcal{L}(\boldsymbol{\theta}_k, \cdot)$. Based on this new convergence metric, we can now introduce the notion of the *approximate first-order stationary points*.

Definition 2. The point $\{\boldsymbol{\theta}, \boldsymbol{\omega}\}$ is an ϵ-stationary point of function $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\omega})$ if $\mathfrak{M}(\boldsymbol{\theta}, \boldsymbol{\omega}) \leq \epsilon$.

Several important remarks on the connections between our metric $\mathfrak{M}^{(k)}$ and the conventional convergence metrics in the literature are in order. A conventional convergence metric in the literature for NCX-SCV minimax optimization is $\|\nabla J(\boldsymbol{\theta}^{(k)})\|^2$ (Lin et al., 2020a; Luo et al., 2020; Zhang et al., 2021; Wu* et al., 2023; Huang et al., 2021; Wu et al., 2023), which is the first term of $\mathfrak{M}^{(k)}$. This is because $\|\nabla J(\boldsymbol{\theta})\|^2 = 0$ implies that $\boldsymbol{\theta}$ is a FOSP. Another conventional convergence metric in the literature of minimizing the empirical MSPBE problem is $\|\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\omega})\|^2 + \|\nabla_{\boldsymbol{\omega}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\omega})\|^2$ (Tsitsiklis & Van Roy, 1996). Since the nonconvex-strong-concave minimax optimization problem is unconstrained in dual (i.e., $\boldsymbol{\omega} \in \mathbb{R}^d$), it follows from Lipschitz-smoothness in Assumption 2 and $\|\nabla_{\boldsymbol{\omega}} \mathcal{L}(\boldsymbol{\theta}^{(k)}, \boldsymbol{\omega}^*(\boldsymbol{\theta}^{(k)}))\|^2 = 0$ that $\|\boldsymbol{\omega}^{(k)} - \boldsymbol{\omega}^*(\boldsymbol{\theta}^{(k)})\|^2 \geq L_f^{-2} \|\nabla_{\boldsymbol{\omega}} \mathcal{L}(\boldsymbol{\theta}^{(k)}, \boldsymbol{\omega}^{(k)})\|^2$. Therefore, the second term in our $\mathfrak{M}^{(k)}$ (i.e., $2\|\boldsymbol{\omega}^{(k)} - \boldsymbol{\omega}^*(\boldsymbol{\theta}^{(k)})\|^2$) is an upper bound of the second term in this conventional metric (i.e., $\|\nabla_{\boldsymbol{\omega}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\omega})\|^2$). Thus, $2\|\boldsymbol{\omega}^{(k)} - \boldsymbol{\omega}^*(\boldsymbol{\theta}^{(k)})\|^2$ is a *stronger* metric than $\|\nabla_{\boldsymbol{\omega}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\omega})\|^2$ in the sense that an $\mathcal{O}(1/K)$ convergence rate under $\mathfrak{M}^{(k)}$ implies an $\mathcal{O}(1/K)$ convergence rate of the conventional metric, but the converse is *not* true. Moreover, the benefit of using $2\|\boldsymbol{\omega}^{(k)} - \boldsymbol{\omega}^*(\boldsymbol{\theta}^{(k)})\|^2$ in our $\mathfrak{M}^{(k)}$ is that its special structure allows us to prove the $\mathcal{O}(1/K)$ convergence, while the second term in the conventional metric does not enjoy such a salient feature.

With our proposed convergence metric in (6), we have the following convergence result:

Theorem 1. Under Assumptions 1–3, choose step-sizes: $\alpha \leq \min\{\frac{1}{4L_f}, \frac{2\mu}{34L_f^2+2\mu^2}\}$ and $\beta \leq \min\left\{\frac{1}{4L_f}, \frac{1}{2(L_f+L_f^2/\mu)}, \frac{\mu}{8\sqrt{17}L_f^2}, \frac{\mu^2\alpha}{8\sqrt{34}L_f^2}\right\}$. Let $q = |\mathcal{N}| = \lceil\sqrt{M}\rceil$, it holds that:

$$\frac{1}{K}\sum_{k=0}^{K-1} \mathbb{E}[\mathfrak{M}^{(k)}] \leq \frac{1}{K\min\{1, L_f^2\}}\left[\frac{16L_f^2}{\alpha\mu}C_2 + \frac{2}{\beta}C_1\right] = \mathcal{O}\left(\frac{1}{K}\right),$$

where $C_1 \triangleq \mathbb{E}[J(\boldsymbol{\theta}^{(0)})] - \mathbb{E}[J(\boldsymbol{\theta}^*)]$, $C_2 \triangleq \left(\mathbb{E}\|\boldsymbol{\omega}^*(\boldsymbol{\theta}^{(0)}) - \boldsymbol{\omega}^{(0)}\|^2\right)$ and $\boldsymbol{\theta}^* = \min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$.

Theorem 1 immediately implies the following result:

Corollary 1. The overall sample complexity of PILOT is $\mathcal{O}(\sqrt{M}\kappa^3\epsilon^{-1} + M)$, where $\kappa = L_f/\mu$ denotes the condition number.

Theorem 1 states that PILOT achieves an $\mathcal{O}(1/K)$ convergence rate to an ϵ-FOSP. The most challenging part in proving Theorem 1 stems from the fact that one needs to simultaneously evaluate the progresses of the gradient descent in the primal domain and the gradient ascent in the dual domain of the minimax problem.

[4] To see this, recall that $\boldsymbol{D} = \mathbb{E}_s\left[\nabla_\theta V_\theta(s) \nabla_\theta V_\theta(s)^\top\right] \in \mathbb{R}^{d\times d}$. Note that $\mu = \lambda_{\min}(\boldsymbol{D}) > 0$ since $\boldsymbol{D}$ is positive definite and $\boldsymbol{D}$ tends to be full-rank as M increases. Thus, as soon as we find a $\mu > 0$ when M is sufficiently large, this μ is independent of M as M continues to increase.

Remark 2. It is worth noting that the nPD-VR method in (Wai et al., 2019) employs $\|\nabla_{\omega}\mathcal{L}(\boldsymbol{\theta}^{(k)},\boldsymbol{\omega}^{(k)})\|^2$ in their metric to evaluate convergence. However, this approach yields a term $F^{(K)} \triangleq \mathbb{E}[\mathcal{L}(\boldsymbol{\theta}^{(0)},\boldsymbol{\omega}^{(0)}) - \mathcal{L}(\boldsymbol{\theta}^{(K)},\boldsymbol{\omega}^{(K)})]$ in their convergence upper bound in the form of $\mathcal{O}(F^{(K)}/K)$ (cf. Theorem 1, Eq. (26) in (Wai et al., 2019)). Since $F^{(K)}$ depends on K, it is unclear whether the nPD-VR method in (Wai et al., 2019) can achieve an $\mathcal{O}(1/K)$ convergence rate or not. This ambiguous result motivates us to propose this new metric $\mathfrak{M}^{(k)}$ in Eq. (6) to evaluate the convergence of our PILOT algorithm. Consequently, we bound per-iteration change in $J(\boldsymbol{\theta})$ instead of the function $\mathcal{L}(\boldsymbol{\theta}^{(k)},\boldsymbol{\omega}^{(k)})$. This helps us avoid the technical limitations of (Wai et al., 2019) and successfully establish the $\mathcal{O}(1/K)$ convergence rate. In addition to the $\mathcal{O}(1/K)$ convergence rate, our PILOT algorithm also enjoys the following salient features:

a) Large and Constant Step-Sizes: It is worth noting that PILOT adopts a large $\mathcal{O}(1)$ (i.e., constant) step-size compared to the $\mathcal{O}(1/M)$ step-size of nPD-VR (Wai et al., 2019), where M represents the dataset size. This also induces a faster empirical convergence speed. Besides, PILOT's estimator uses fresher information from the previous iteration (see Feature c) below), while VR-STSG (Qiu et al., 2020) and nPD-VR (Wai et al., 2019) only use the information from the beginning of q-sized windows. Collectively, PILOT makes considerably larger progress than state-of-the-art algorithms (Qiu et al., 2020; Wai et al., 2019).

b) A Recursive Path-Following VR Approach for minimax Problems: In the literature, most existing single-timescale methods adopt vanilla stochastic gradients as gradient estimators $G_{\theta,t}$ and $G_{\omega,t}$, which suffer from slow convergence rates. To the best of our knowledge, the only VR-based single-timescale method is (Qiu et al., 2020). However, (Qiu et al., 2020) is based on the SVRG-type VR technique, which achieves a slower $\mathcal{O}(1/K^{3/2})$ convergence rate. In comparison, our work is based on an advanced recursive path-following VR-based update, which enables the use of constant step-sizes to achieve the first $\mathcal{O}(1/K)$ convergence rate in the literature.

5 The Adaptive-Batch PILOT Method (PILOT$^+$)

Note that PILOT still requires full gradients every q iterations. This motivates us to propose an adaptive-batch method called PILOT$^+$ to further lower the sample complexity.

1) Algorithm Description: The full description of PILOT$^+$ is illustrated in Algorithm 2. In PILOT$^+$, our key idea is to use the gradients calculated in the previous loop to adjust the batch size $\mathcal{N}_s$ of the next loop. Specifically, PILOT$^+$ chooses $\mathcal{N}_s$ at the k-th iteration as:

$$|\mathcal{N}_s| = \min\{c_\gamma\sigma^2(\gamma^{(k)})^{-1}, c_\epsilon\sigma^2\epsilon^{-1}, M\}, \tag{7}$$

where $c_\gamma, c_\epsilon > c$ for certain constant c, M denotes the size of the dataset, and $\gamma^{(k+1)} = \sum_{i=(n_k-1)q}^{k}\|G_{\boldsymbol{\theta}}^{(i)}\|^2/q$ is the stochastic gradients calculated in the previous iterations. In PILOT$^+$, for every q iterations, we select $\mathcal{N}_s$ samples independently and uniformly from $[M]$ and compute gradient estimators as follows:

$$G_{\boldsymbol{\theta}}^{(k)} = \frac{1}{|\mathcal{N}_s|}\sum_{i\in\mathcal{N}_s}\nabla_{\boldsymbol{\theta}}\mathcal{L}_i(\boldsymbol{\theta}^{(k)},\boldsymbol{\omega}^{(k)}), \quad G_{\boldsymbol{\omega}}^{(k)} = \frac{1}{|\mathcal{N}_s|}\sum_{i\in\mathcal{N}_s}\nabla_{\boldsymbol{\omega}}\mathcal{L}_i(\boldsymbol{\theta}^{(k)},\boldsymbol{\omega}^{(k)}). \tag{8}$$

For other iterations, PILOT$^+$ is exactly the same as PILOT. Next, we will theoretically show that such an adaptive batch-size scheme still retains the same convergence rate, while achieving an improved sample complexity.

2) Convergence Performance: For PILOT$^+$, we have the following theoretical convergence result:

Theorem 2. Under Assumptions 1–4, choose step-sizes: $\alpha \le \min\{\frac{1}{4L_f}, \frac{2\mu}{34L_f^2+2\mu^2}\}$ and $\beta \le \min\{\frac{1}{4L_f}, \frac{1}{2(L_f+L_f^2/\mu)}, \frac{\mu}{8\sqrt{17}L_f^2}, \frac{\mu^2\alpha}{8\sqrt{34}L_f^2}\}$. Let $|\mathcal{N}_s| = \min\{c_\gamma\sigma^2(\gamma^{(k)})^{-1}, c_\epsilon\sigma^2\epsilon^{-1}, M\}$, $q = \lceil\sqrt{M}\rceil$, $|\mathcal{N}| = \lceil\sqrt{M}\rceil$ and $c_\gamma \ge (288L_f^2/\mu^2 + 8)$ in PILOT$^+$, where $c_\gamma \ge c$ for some constant $c > 4K + \frac{68K}{\beta\mu^2}$. With constants $C_1 \triangleq \mathbb{E}[J(\boldsymbol{\theta}^{(0)})] - \mathbb{E}[J(\boldsymbol{\theta}^*)]$ and $C_2 \triangleq \big(\mathbb{E}[\|\boldsymbol{\omega}^*(\boldsymbol{\theta}^{(0)}) - \boldsymbol{\omega}^{(0)}\|^2]\big)$, it holds that:

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}[\mathfrak{M}^{(k)}] \le \frac{1}{K\min\{1,L_f^2\}}\cdot\left[K\cdot\frac{\epsilon}{2} + \frac{16L_f^2}{\alpha\mu}C_2 + \frac{2}{\beta}C_1\right] = \mathcal{O}\left(\frac{1}{K}\right) + \frac{\epsilon}{2}.$$

Theorem 2 immediately implies the following result:

Algorithm 2 Adaptive-batch PILOT method (PILOT$^+$).

Input: A trajectory of the state-action pairs $\{s_1, a_1, s_2, a_2, \cdots, s_M, a_M, s_{M+1}\}$ generated from a given policy; step sizes $\alpha, \beta \geq 0$; initialization points $\boldsymbol{\theta}^0 \in \Theta$, $\boldsymbol{\omega}^0 \in \mathbb{R}^d$.

Output: $(\boldsymbol{\theta}^{(\widetilde{K})}, \boldsymbol{\omega}^{(\widetilde{K})})$, where $\widetilde{K}$ is independently and uniformly picked from $\{1, \cdots, K\}$;

1: **for** $k = 0, 1, 2, \cdots, K-1$ **do**
2: If $\text{mod}(k, q) = 0$, select $|\mathcal{N}_s|$ indices independently and uniformly from $[M]$ as in Eq. (7) and calculate stochastic gradients as in Eq. (8);
3: If $\text{mod}(k, q) \neq 0$, select $|\mathcal{N}|$ samples independently and uniformly from $[M]$; Compute gradients as in Eq. (4);
4: Perform the primal-dual updates as in Eq. (5).
5: **end for**

Corollary 2. The overall sample complexity of PILOT$^+$ is $\mathcal{O}(\sqrt{M}\kappa^3\epsilon^{-1} + M)$, where $\kappa = L_f/\mu$ denotes the condition number.

6 Experimental Results

In this section, we conduct our numerical experiments to verify our theoretical results. We compare our work with the basic stochastic gradient (SG) method (Lin et al., 2020b) and three state-of-the-art algorithms for PE: nPD-VR (Wai et al., 2019), STSG (Qiu et al., 2020) and VR-STSG (Qiu et al., 2020). Due to space limitation, we provide our detailed experiment settings in the Appendix.

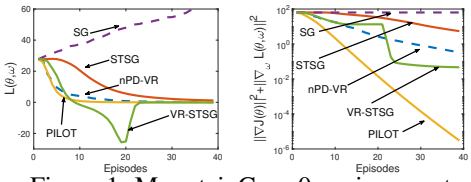

Figure 1: MountainCar-v0 environment.

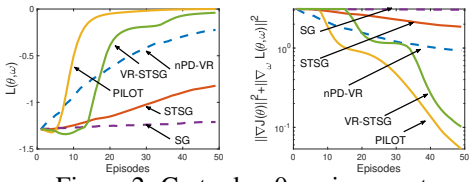

Figure 2: Cartpole-v0 environment.

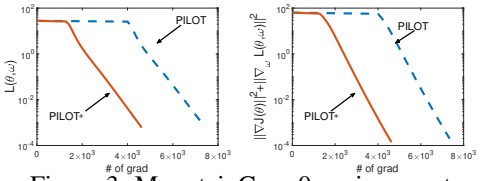

Figure 3: MountainCar-v0 environment.

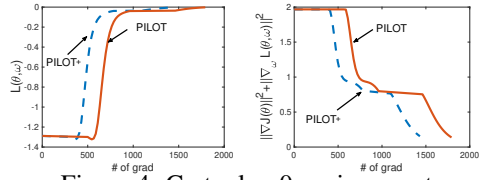

Figure 4: Cartpole-v0 environment.

Numerical Results: First, we compare the loss value and gradient norm performance based on MountainCar-v0 and Cartpole-v0 with nPD-VR, SG, STSG, and VR-STSG in Figs. 1 and 2. We initialize all algorithms at the same point, which is generated randomly from the normal distribution. We can see that VR-STSG and nPD-VR slowly converge after 40 epochs, while STSG and SG fail to converge. PILOT converges faster than all the other algorithms with the same step-size values. As for Cartpole-v0, we clearly see a trend of approaching zero-loss with PILOT. These results are consistent with our theoretical result that one can use a relatively large step-size with PILOT, which leads to a faster convergence performance. Also, we compare the sample complexity of PILOT and PILOT$^+$ in MountainCar-v0 and Cartpole-v0, and the results are shown in in Figs. 3 and 4, respectively. We can see that PILOT$^+$ converges to the same level with much fewer samples than PILOT does.

7 Conclusion

In this paper, we proposed two algorithms called PILOT and PILOT$^+$ for PE with nonlinear approximation and performed the theoretical analysis of their convergence and sample complexity. The PILOT algorithm is based on a single-timescale framework by utilizing VR techniques. The PILOT algorithm allows the use of constant step-sizes and achieves an $\mathcal{O}(1/K)$ convergence rate. The PILOT$^+$ algorithm improves the sample complexity of PILOT by further applying an adaptive batch size based on historical stochastic gradient information. Our experimental results also confirmed our theoretical findings in convergence and sample complexity.

ACKNOWLEDGMENTS AND DISCLOSURE OF FUNDING

This work has been supported in part by NSF grants CAREER CNS-2110259 and CNS-2112471.

REFERENCES

Justin A Boyan. Technical update: Least-squares temporal difference learning. *Machine Learning*, 49:233–246, 2002.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Tianyi Chen, Kaiqing Zhang, Georgios B Giannakis, and Tamer Basar. Communication-efficient policy gradient methods for distributed reinforcement learning. *IEEE Transactions on Control of Network Systems*, 9(2):917–929, 2021.

Wesley Chung, Somjit Nath, Ajin Joseph, and Martha White. Two-timescale networks for nonlinear value function approximation. In *International Conference on Learning Representations*, 2018.

Gal Dalal, Gugan Thoppe, Balázs Szörényi, and Shie Mannor. Finite sample analysis of two-timescale stochastic approximation with applications to reinforcement learning. In *Conference on Learning Theory*, pp. 1199–1233. PMLR, 2018.

Christoph Dann, Gerhard Neumann, Jan Peters, et al. Policy evaluation with temporal differences: A survey and comparison. *Journal of Machine Learning Research*, 15:809–883, 2014.

Constantinos Daskalakis, Stratis Skoulakis, and Manolis Zampetakis. The complexity of constrained min-max optimization. In *the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 1466–1478, 2021.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in Neural Information Processing Systems*, 27, 2014.

Thinh Doan, Siva Maguluri, and Justin Romberg. Finite-time analysis of distributed td (0) with linear function approximation on multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 1626–1635. PMLR, 2019.

Simon S Du, Jianshu Chen, Lihong Li, Lin Xiao, and Dengyong Zhou. Stochastic variance reduction methods for policy evaluation. In *International Conference on Machine Learning*, pp. 1049–1058. PMLR, 2017.

Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. *Advances in Neural Information Processing Systems*, 31, 2018.

Feihu Huang, Xidong Wu, and Heng Huang. Efficient mirror descent ascent methods for nonsmooth minimax problems. *NeurIPS 21:Advances in Neural Information Processing Systems*, 34:10431–10443, 2021.

Kaiyi Ji, Zhe Wang, Bowen Weng, Yi Zhou, Wei Zhang, and Yingbin Liang. History-gradient aided batch size adaptation for variance reduced algorithms. In *International Conference on Machine Learning*, pp. 4762–4772. PMLR, 2020.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in Neural Information Processing Systems*, 26, 2013.

Michail G Lagoudakis and Ronald Parr. Least-squares policy iteration. *Journal of Machine Learning Research*, 4(Dec):1107–1149, 2003.

Lihua Lei and Michael I Jordan. On the adaptivity of stochastic gradient-based optimization. *SIAM Journal on Optimization*, 30(2):1473–1500, 2020.

Lihua Lei, Cheng Ju, Jianbo Chen, and Michael I Jordan. Non-convex finite-sum optimization via scsg methods. *Advances in Neural Information Processing Systems*, 30, 2017.

Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

Tianjiao Li, Guanghui Lan, and Ashwin Pananjady. Accelerated and instance-optimal policy evaluation with linear function approximation. *arXiv preprint arXiv:2112.13109*, 2021.

Xiangru Lian, Mengdi Wang, and Ji Liu. Finite-sum composition optimization via variance reduced gradient descent. In *Artificial Intelligence and Statistics*, pp. 1159–1167. PMLR, 2017.

Tianyi Lin, Chengyou Fan, Mengdi Wang, and Michael I Jordan. Improved sample complexity for stochastic compositional variance reduced gradient. In *American Control Conference (ACC)*, pp. 126–131. IEEE, 2020a.

Tianyi Lin, Chi Jin, and Michael Jordan. On gradient descent ascent for nonconvex-concave minimax problems. In *International Conference on Machine Learning*, pp. 6083–6093. PMLR, 2020b.

Bo Liu, Ji Liu, Mohammad Ghavamzadeh, Sridhar Mahadevan, and Marek Petrik. Finite-sample analysis of proximal gradient td algorithms. In *Uncertainty in Artificial Intelligence*, pp. 504–513, 2015.

Bo Liu, Ian Gemp, Mohammad Ghavamzadeh, Ji Liu, Sridhar Mahadevan, and Marek Petrik. Proximal gradient temporal difference learning: Stable reinforcement learning with polynomial sample complexity. *Journal of Artificial Intelligence Research*, 63:461–494, 2018.

Songtao Lu, Ioannis Tsaknakis, and Mingyi Hong. Block alternating optimization for non-convex min-max problems: algorithms and applications in signal processing and communications. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4754–4758, 2019.

Songtao Lu, Ioannis Tsaknakis, Mingyi Hong, and Yongxin Chen. Hybrid block successive approximation for one-sided non-convex min-max problems: algorithms and applications. *IEEE Transactions on Signal Processing*, 68:3676–3691, 2020.

Luo Luo, Haishan Ye, Zhichao Huang, and Tong Zhang. Stochastic recursive gradient descent ascent for stochastic nonconvex-strongly-concave minimax problems. *Advances in Neural Information Processing Systems*, 33, 2020.

Sergio Valcarcel Macua, Jianshu Chen, Santiago Zazo, and Ali H Sayed. Distributed policy evaluation under multiple behavior strategies. *IEEE Transactions on Automatic Control*, 60(5):1260–1274, 2014.

Hamid Maei, Csaba Szepesvari, Shalabh Bhatnagar, Doina Precup, David Silver, and Richard S Sutton. Convergent temporal-difference learning with arbitrary smooth function approximation. *Advances in Neural Information Processing Systems*, 22, 2009.

Oren Mangoubi and Nisheeth K Vishnoi. Greedy adversarial equilibrium: an efficient alternative to nonconvex-nonconcave min-max optimization. In *the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 896–909, 2021.

Hongzi Mao, Mohammad Alizadeh, Ishai Menache, and Srikanth Kandula. Resource management with deep reinforcement learning. In *the 15th ACM Workshop on Hot Topics in Networks*, pp. 50–56, 2016.

Yurii Nesterov et al. *Lectures on convex optimization*, volume 137. Springer, 2018.

Maher Nouiehed, Maziar Sanjabi, Tianjian Huang, Jason D Lee, and Meisam Razaviyayn. Solving a class of non-convex min-max games using iterative first order methods. *Advances in Neural Information Processing Systems*, 32, 2019.

Gandharv Patil, LA Prashanth, Dheeraj Nagaraj, and Doina Precup. Finite time analysis of temporal difference learning with linear function approximation: Tail averaging and regularisation. In *International Conference on Artificial Intelligence and Statistics*, pp. 5438–5448. PMLR, 2023.

Brenden K Petersen, Jiachen Yang, Will S Grathwohl, Chase Cockrell, Claudio Santiago, Gary An, and Daniel M Faissol. Deep reinforcement learning and simulation as a path toward precision medicine. *Journal of Computational Biology*, 26(6):597–604, 2019.

Shuang Qiu, Zhuoran Yang, Xiaohan Wei, Jieping Ye, and Zhaoran Wang. Single-timescale stochastic nonconvex-concave optimization for smooth nonlinear td learning. *arXiv preprint arXiv:2008.10103*, 2020.

Aniruddh Raghu, Matthieu Komorowski, Imran Ahmed, Leo Celi, Peter Szolovits, and Marzyeh Ghassemi. Deep reinforcement learning for sepsis treatment. *arXiv preprint arXiv:1711.09602*, 2017a.

Aniruddh Raghu, Matthieu Komorowski, Leo Anthony Celi, Peter Szolovits, and Marzyeh Ghassemi. Continuous state-space models for optimal sepsis treatment: a deep reinforcement learning approach. In *Machine Learning for Healthcare Conference*, pp. 147–163. PMLR, 2017b.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pp. 400–407, 1951.

Dan Simon. A game theory approach to constrained minimax state estimation. *IEEE Transactions on Signal Processing*, 54(2):405–412, 2006.

Chaobing Song, Zhengyuan Zhou, Yichao Zhou, Yong Jiang, and Yi Ma. Optimistic dual extrapolation for coherent non-monotone variational inequalities. *Advances in Neural Information Processing Systems*, 33:14303–14314, 2020.

Rayadurgam Srikant and Lei Ying. Finite-time error bounds for linear stochastic approximation andtd learning. In *Conference on Learning Theory*, pp. 2803–2830. PMLR, 2019.

Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3 (1):9–44, 1988.

Richard S Sutton, Csaba Szepesvári, and Hamid Reza Maei. A convergent $o(n)$ algorithm for off-policy temporal-difference learning with linear function approximation. *Advances in Neural Information Processing Systems*, 21, 2008.

Richard S Sutton, Hamid Reza Maei, Doina Precup, Shalabh Bhatnagar, David Silver, Csaba Szepesvári, and Eric Wiewiora. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *the 26th Annual International Conference on Machine Learning*, pp. 993–1000, 2009.

Georgios Theocharous, Philip S Thomas, and Mohammad Ghavamzadeh. Personalized ad recommendation systems for life-time value optimization with guarantees. In *the 24th International Joint Conference on Artificial Intelligence*, 2015.

Ahmed Touati, Pierre-Luc Bacon, Doina Precup, and Pascal Vincent. Convergent tree backup and retrace with function approximation. In *International Conference on Machine Learning*, pp. 4955–4964. PMLR, 2018.

John Tsitsiklis and Benjamin Van Roy. Analysis of temporal-diffference learning with function approximation. *Advances in Neural Information Processing Systems*, 9, 1996.

Hoi-To Wai, Mingyi Hong, Zhuoran Yang, Zhaoran Wang, and Kexin Tang. Variance reduced policy evaluation with smooth function approximation. *Advances in Neural Information Processing Systems*, 32, 2019.

Mengdi Wang, Ji Liu, and Ethan Fang. Accelerating stochastic composition optimization. *Advances in Neural Information Processing Systems*, 29, 2016.

Mengdi Wang, Ethan X Fang, and Han Liu. Stochastic compositional gradient descent: algorithms for minimizing compositions of expected-value functions. *Mathematical Programming*, 161(1-2): 419–449, 2017.

Zhe Wang, Kaiyi Ji, Yi Zhou, Yingbin Liang, and Vahid Tarokh. Spiderboost and momentum: Faster variance reduction algorithms. *Advances in Neural Information Processing Systems*, 32, 2019.

Wei Wen, Kuang-Hung Liu, Igor Fedorov, Xin Zhang, Hang Yin, Weiwei Chu, Kaveh Hassani, Mengying Sun, Jiang Liu, Xu Wang, et al. Rankitect: Ranking architecture search battling world-class engineers at meta scale. *arXiv preprint arXiv:2311.08430*, 2023.

Xidong Wu, Zhengmian Hu, and Heng Huang. Decentralized riemannian algorithm for nonconvex minimax problems. In *AAAI 23: Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

Xidong Wu*, Jianhui Sun*, Zhengmian Hu, Aidong Zhang, and Heng Huang. Solving a class of non-convex minimax optimization in federated learning. *NeurIPS 23: Advances in Neural Information Processing Systems*, 2023.

Tengyu Xu, Zhe Wang, Yi Zhou, and Yingbin Liang. Reanalysis of variance reduced temporal difference learning. In *International Conference on Learning Representations*, 2019.

Tengyu Xu, Zhe Wang, Yingbin Liang, and H. Vincent Poor. Enhanced first and zeroth order variance reduced algorithms for min-max optimization. *ArXiv*, abs/2006.09361, 2020. URL `https://api.semanticscholar.org/CorpusID:219708545`.

Junyu Zhang and Lin Xiao. A stochastic composite gradient method with incremental variance reduction. *Advances in Neural Information Processing Systems*, 32, 2019.

Siqi Zhang, Junchi Yang, Cristóbal Guzmán, Negar Kiyavash, and Niao He. The complexity of nonconvex-strongly-concave minimax optimization. In *Uncertainty in Artificial Intelligence*, pp. 482–492. PMLR, 2021.

Xingyu Zhou. On the fenchel duality between strong convexity and lipschitz continuous gradient. *arXiv preprint arXiv:1803.06573*, 2018.