SBEED: Convergent Reinforcement Learning with Nonlinear Function Approximation

Bo Dai [1] **Albert Shaw** [1] **Lihong Li** [2] **Lin Xiao** [3] **Niao He** [4] **Zhen Liu** [1] **Jianshu Chen** [5] **Le Song** [1]

Abstract

When function approximation is used, solving the Bellman optimality equation with stability guarantees has remained a major open problem in reinforcement learning for decades. The fundamental difficulty is that the Bellman operator may become an expansion in general, resulting in oscillating and even divergent behavior of popular algorithms like Q-learning. In this paper, we revisit the Bellman equation, and reformulate it into a novel primal-dual optimization problem using Nesterov's smoothing technique and the Legendre-Fenchel transformation. We then develop a new algorithm, called *Smoothed Bellman Error Embedding*, to solve this optimization problem where any differentiable function class may be used. We provide what we believe to be the first convergence guarantee for general nonlinear function approximation, and analyze the algorithm's sample complexity. Empirically, our algorithm compares favorably to state-of-the-art baselines in several benchmark control problems.

1. Introduction

In reinforcement learning (RL), the goal of an agent is to learn a policy that maximizes long-term returns by sequentially interacting with an unknown environment (Sutton & Barto, 1998). The dominating framework to model such an interaction is the Markov decision process, or MDP, in which the optimal value function are characterized as a fixed point of the Bellman operator. A fundamental result for MDP is that the Bellman operator is a contraction in the value-function space, so the optimal value function is the unique fixed point. Furthermore, starting from any initial value function, iterative applications of the Bellman operator ensure convergence to the fixed point. Interested readers are referred to the textbook of Puterman (2014) for details.

Many of the most effective RL algorithms have their root in such a fixed-point view. The most prominent family of algorithms is perhaps the temporal-difference algorithms, including TD(λ) (Sutton, 1988), Q-learning (Watkins, 1989), SARSA (Rummery & Niranjan, 1994; Sutton, 1996), and numerous variants such as the empirically very successful DQN (Mnih et al., 2015) and A3C (Mnih et al., 2016) implementations. Compared to direct policy search/gradient algorithms like REINFORCE (Williams, 1992), these fixed-point methods make learning more efficient by *bootstrapping* (a sample-based version of Bellman operator).

When the Bellman operator can be computed exactly (even on average), such as when the MDP has finite state/actions, convergence is guaranteed thanks to the contraction property (Bertsekas & Tsitsiklis, 1996). Unfortunately, when function approximatiors are used, such fixed-point methods *easily* become unstable or even divergent (Boyan & Moore, 1995; Baird, 1995; Tsitsiklis & Van Roy, 1997), except in a few special cases. For example,

- for some rather restrictive function classes, such as those with a non-expansion property, some of the finite-state MDP theory continues to apply with proper modifications (Gordon, 1995; Ormoneit & Sen, 2002; Antos et al., 2008);
- when *linear* value function approximation in certain cases, convergence is guaranteed: for evaluating a *fixed* policy from *on-policy* samples (Tsitsiklis & Van Roy, 1997), for evaluating the policy using a closed-form solution from *off-policy* samples (Boyan, 2002; Lagoudakis & Parr, 2003), or for optimizing a policy using samples collected by a stationary policy (Maei et al., 2010).

In recent years, a few authors have made important progress toward finding scalable, convergent TD algorithms, by designing proper objective functions and using stochastic gradient descent (SGD) to optimize them (Sutton et al., 2009; Maei, 2011). Later on, it was realized that several of these gradient-based algorithms can be interpreted as solving a primal-dual problem (Mahadevan et al., 2014; Liu et al., 2015; Macua et al., 2015; Dai et al., 2017). This insight has

[1]Georgia Institute of Technology [2]Google Inc. [3]Microsoft Research [4]University of Illinois at Urbana Champaign [5]Tencent AI Lab. Correspondence to: Bo Dai <bodai@gatech.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

led to novel, faster, and more robust algorithms by adopting sophisticated optimization techniques (Du et al., 2017). Unfortunately, to the best of our knowledge, all existing works either assume linear function approximation or are designed for policy evaluation. It remains a major open problem how to find the *optimal policy* reliably with general *nonlinear* function approximators such as neural networks, especially in the presence of *off-policy* data.

Contributions In this work, we take a substantial step towards solving this decades-long open problem, leveraging a powerful saddle-point optimization perspective, to derive a new algorithm called *Smoothed Bellman Error Embedding (SBEED) algorithm*. Our development hinges upon a novel view of a smoothed Bellman optimality equation, which is then transformed to the final primal-dual optimization problem. SBEED learns the optimal value function and a stochstic policy in the primal, and the Bellman error (also known as Bellman residual) in the dual. By doing so, it avoids the non-smooth $\max$-operator in the Bellman operator, as well as the double-sample challenge that has plagued RL algorithm designs (Baird, 1995). More specifically,

- SBEED is stable for a broad class of nonlinear function approximators including neural networks, and provably converges to a solution with vanishing gradient. This holds even in the more challenging off-policy case;
- it uses bootstrapping to yield high sample efficiency, as in TD-style methods, and is also generalized to cases of multi-step bootstrapping and eligibility traces;
- it avoids the double-sample issue and directly optimizes the squared Bellman error based on sample trajectories;
- it uses stochastic gradient descent to optimize the objective, thus very efficient and scalable.

Furthermore, the algorithm handles both the optimal value function estimation and policy optimization in a unified way, and readily applies to both continuous and discrete action spaces. We compare the algorithm with state-of-the-art baselines on several continuous control benchmarks, and obtain excellent results.

2. Preliminaries

In this section, we introduce notation and technical background that is needed in the rest of the paper. We denote a Markov decision process (MDP) as $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$, where $\mathcal{S}$ is a (possible infinite) state space, $\mathcal{A}$ an action space, $P(\cdot|s,a)$ the transition probability kernel defining the distribution over next states upon taking action a on state s, $R(s,a)$ the average immediate reward by taking action a in state s, and $\gamma \in (0,1)$ a discount factor. Given an MDP, we wish to find a possibly stochastic policy $\pi : \mathcal{S} \to \mathcal{P}_{\mathcal{A}}$ to maximize the expected discounted cumulative reward starting from any state $s \in \mathcal{S}$: $\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \middle| s_0 = s, \pi\right]$, where $\mathcal{P}_{\mathcal{A}}$ denotes all probability measures over $\mathcal{A}$. The set of all policies is denoted by $\mathcal{P} := (\mathcal{P}_{\mathcal{A}})^{\mathcal{S}}$.

Define $V^*(s) := \max_{\pi(\cdot|s)} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) | s_0 = s, \pi\right]$ to be the optimal value function. It is known that V^* is the unique fixed point of the Bellman operator $\mathcal{T}$, or equivalently, the unique solution to the Bellman optimality equation (Bellman equation, for short) (Puterman, 2014):

$$V(s) = (\mathcal{T}V)(s) := \max_a R(s,a) + \gamma \mathbb{E}_{s'|s,a}\left[V(s')\right]. \quad (1)$$

The optimal policy π^* is related to V^* by the following:

$$\pi^*(a|s) = \operatorname*{argmax}_a \left\{R(s,a) + \gamma \mathbb{E}_{s'|s,a}\left[V^*(s')\right]\right\}.$$

It should be noted that in practice, for convenience we often work on the Q-function instead of the state-value function V^*. In this paper, it suffices to use the simpler V^* function.

3. A Primal-Dual View of Bellman Equation

In this section, we introduce a novel view of Bellman equation that enables the development of the new algorithm in Section 4. After reviewing the Bellman equation and the challenges to solve it, we describe the two key technical ingredients that lead to our primal-dual reformulation.

We start with another version of Bellman equation that is equivalent to Eqn (1) (see, *e.g.*, Puterman (2014)):

$$V(s) = \max_{\pi(\cdot|s) \in \mathcal{P}_{\mathcal{A}}} \mathbb{E}_{a \sim \pi(\cdot|s)}\left[R(s,a) + \gamma \mathbb{E}_{s'|s,a}\left[V(s')\right]\right]. \quad (2)$$

Eqn (2) makes the role of a policy explicit. Naturally, one may try to jointly optimize over V and π to minimize the discrepancy between the two sides of (2). For concreteness, we focus on the square distance in this paper, but our results can be extended to other convex loss functions. Let μ be some given state distribution so that $\mu(s) > 0$ for all $s \in \mathcal{S}$. Minimizing the *squared Bellman error* gives the following:

$$\min_V \mathbb{E}_{s \sim \mu}\Big[\big(\max_{\pi(\cdot|s) \in \mathcal{P}_{\mathcal{A}}} \mathbb{E}_{a \sim \pi(\cdot|s)}\big[R(s,a) + \gamma \mathbb{E}_{s'|s,a}\left[V(s')\right]\big] - V(s)\big)^2\Big]. \quad (3)$$

While being natural, this approach has several major difficulties when it comes to optimization, which are to be dealt with in the following subsections:

1. The $\max$ operator over $\mathcal{P}_{\mathcal{A}}$ introduces non-smoothness to the objective function. A slight change in V may cause large differences in the RHS of Eqn (2).
2. The conditional expectation, $\mathbb{E}_{s'|s,a}[\cdot]$, composed with the square loss, requires double samples (Baird, 1995) to obtain unbiased gradients, which is often impractical in most but simulated environments.

3.1. Smoothed Bellman Equation

To avoid the instability and discontinuity caused by the $\max$ operator, we use the smoothing technique of Nesterov

(2005) to smooth the Bellman operator $\mathcal{T}$. Since policies are conditional distributions over $\mathcal{A}$, we choose entropy regularization, and Eqn (2) becomes:

$$V_\lambda(s) = \max_{\pi(\cdot|s)\in\mathcal{P}_\mathcal{A}} \Big(\mathbb{E}_{a\sim\pi(\cdot|s)}\big(R(s,a) + \gamma\mathbb{E}_{s'|s,a}\left[V_\lambda(s')\right]\big) + \lambda H(\pi,s)\Big), \quad (4)$$

where $H(\pi,s) := -\sum_{a\in\mathcal{A}}\pi(a|s)\log\pi(a|s)$, and $\lambda \geqslant 0$ controls the degree of smoothing. Note that with $\lambda = 0$, we obtain the standard Bellman equation. Moreover, the regularization may be viewed a *shaping* reward added to the reward function of an induced, equivalent MDP; see the long version of this paper for more details[1].

Since negative entropy is the conjugate of the log-sum-exp function (Boyd & Vandenberghe, 2004, Example 3.25), Eqn (4) can be written equivalently as

$$V_\lambda(s) = \left(\mathcal{T}_\lambda V_\lambda\right)(s) \quad (5)$$
$$:= \lambda\log\left(\sum_{a\in\mathcal{A}}\exp\left(\frac{R(s,a)+\gamma\mathbb{E}_{s'|s,a}\left[V_\lambda(s')\right]}{\lambda}\right)\right),$$

where the log-sum-exp is an effective smoothing approximation of the max-operator.

Remark. While Eqns (4) and (5) are inspired by Nestorov smoothing technique, they can also be derived from other principles (Rawlik et al., 2012; Fox et al., 2016; Neu et al., 2017; Nachum et al., 2017; Asadi & Littman, 2017). For example, Nachum et al. (2017) propose PCL which use entropy regularization in the policy space to encourage exploration, but arrive at the same smoothed form; the smoothed operator $\mathcal{T}_\lambda$ is called "Mellowmax" by Asadi & Littman (2017), which is obtained as a particular instantiation of the quasi-arithmetic mean. In the rest of the subsection, we review the properties of $\mathcal{T}_\lambda$, although some of the results have appeared in the literature in slightly different forms.

First, we show $\mathcal{T}_\lambda$ is also a contraction, as with the standard Bellman operator (Fox et al., 2016; Asadi & Littman, 2017):

Proposition 1 (Contraction) *$\mathcal{T}_\lambda$ is a γ-contraction. Consequently, the corresponding smoothed Bellman equation (4), or equivalently (5), has a unique solution V^*_λ.*

Second, we show that while in general $V^* \neq V^*_\lambda$, their difference is controlled by λ. To do so, define $H^* := \max_{s\in\mathcal{S},\pi(\cdot|s)\in\mathcal{P}_\mathcal{A}} H(\pi,s)$. For finite action spaces, we immediately have $H^* = \log(|\mathcal{A}|)$.

Proposition 2 (Smoothing bias) *Let V^* and V^*_λ be fixed points of (2) and (4), respectively. Then,*

$$\left\|V^*(s) - V^*_\lambda(s)\right\|_\infty \leqslant \frac{\lambda H^*}{1-\gamma}.$$

Consequently, as $\lambda \to 0$, V^*_λ converges to V^* pointwisely.

Finally, the smoothed Bellman operator has the very nice property of temporal consistency (Rawlik et al., 2012; Nachum et al., 2017):

Proposition 3 (Temporal consistency) *Assume $\lambda > 0$. Let V^*_λ be the fixed point of (4) and π^*_λ the corresponding policy that attains the maximum on the RHS of (4). Then, $(V^*_\lambda, \pi^*_\lambda)$ is the unique (V,π) pair that satisfies the following equality for all $(s,a)\in\mathcal{S}\times\mathcal{A}$:*

$$V(s) = R(s,a) + \gamma\mathbb{E}_{s'|s,a}\left[V(s')\right] - \lambda\log\pi(a|s). \quad (6)$$

In other words, Eqn (6) provides an easy-to-check condition to characterize the optimal value function and optimal policy on *arbitrary* pair of (s,a), therefore, which is easy to incorporate *off-policy* data. It can also be extended to the multi-step or eligibility-traces cases as in the long version. Later, this condition will be one of the critical foundations to develop our new algorithm.

3.2. Bellman Error Embedding

A natural objective function inspired by (6) is the *mean squared consistency Bellman error*, given by:

$$\min_{V,\pi\in\mathcal{P}} \ell(V,\pi) := \mathbb{E}_{s,a}\Big[\big(R(s,a) + \gamma\mathbb{E}_{s'|s,a}\left[V(s')\right] - \lambda\log\pi(a|s) - V(s)\big)^2\Big], \quad (7)$$

where $\mathbb{E}_{s,a}[\cdot]$ is shorthand for $\mathbb{E}_{s\sim\mu(\cdot),a\sim\pi_b(\cdot|s)}[\cdot]$. Unfortunately, due to the inner conditional expectation, it would require two independent samples of s' (starting from the same (s,a)) to obtain an unbiased estimate of gradient of f, a problem known as the double-sample issue (Baird, 1995). In practice, however, one can rarely obtain two independent samples except in simulated environments.

To bypass this problem, we make use of the conjugate of the square function (Boyd & Vandenberghe, 2004): $x^2 = \max_\nu\left(2\nu x - \nu^2\right)$, as well as the interchangeability principle (Shapiro et al., 2009; Dai et al., 2017) to rewrite the optimization problem (7) into an equivalent form:

$$\min_{V,\pi\in\mathcal{P}}\max_{\nu\in\mathcal{F}_{\mathcal{S}\times\mathcal{A}}} L(V,\pi;\nu) := 2\mathbb{E}_{s,a,s'}\Big[\nu(s,a)\big(R(s,a) + \gamma V(s') - \lambda\log\pi(a|s) - V(s)\big)\Big] - \mathbb{E}_{s,a,s'}\left[\nu^2(s,a)\right], \quad (8)$$

where $\mathcal{F}_{\mathcal{S}\times\mathcal{A}}$ is the set of real-valued functions on $\mathcal{S}\times\mathcal{A}$, $\mathbb{E}_{s,a,s'}[\cdot]$ is shorthand for $\mathbb{E}_{s\sim\mu(\cdot),a\sim\pi_b(\cdot|s),s'\sim P(\cdot|s,a)}[\cdot]$. Note that (8) is not a standard convex-concave saddle-point problem: the objective is convex in V for any fixed (π,ν), and concave in ν for any fixed (V,π), but not necessarily convex in $\pi\in\mathcal{P}$ for any fixed (V,ν).

Remark. In contrast to our saddle-point formulation (8), Nachum et al. (2017) get around the double-sample obstacle by minimizing an upper bound of $\ell(V,\pi)$: $\tilde{\ell}(V,\pi) := \mathbb{E}_{s,a,s'}\left[\left(R(s,a) + \gamma V(s') - \lambda\log\pi(a|s) - V(s)\right)^2\right]$. As is known (Baird, 1995), the gradient of $\tilde{\ell}$ is different from that of f, as it has a conditional variance term coming from the stochastic outcome s'. In problems where this variance is highly heterogeneous across different (s,a) pairs, impact of such a bias can be substantial.

[1] Proofs of the theorems in the paper as well as further details and extensions of the SBEED algorithm are available in the long version at https://arxiv.org/abs/1712.10285 .

Finally, substituting the dual function $\nu(s,a) = \rho(s,a) - V(s)$, the objective in the saddle-point problem becomes

$$\min_{V,\pi} \max_{\rho \in \mathcal{F}_{\mathcal{S}\times\mathcal{A}}} L_1(V,\pi;\rho) \quad := \quad \mathbb{E}_{s,a,s'}\left[\left(\delta(s,a,s') - V(s)\right)^2\right] - \mathbb{E}_{s,a,s'}\left[\left(\delta(s,a,s') - \rho(s,a)\right)^2\right], \quad (9)$$

where $\delta(s,a,s') := R(s,a) + \gamma V(s') - \lambda \log \pi(a|s)$. Note that the first term is $\bar{\ell}(V,\pi)$, the objective used by PCL, and the second term will cancel the extra variance term, which is rigorously proved in our long version. The use of an auxiliary function to cancel the variance is also observed by Antos et al. (2008). On the other hand, when function approximation is used, extra bias will also be introduced. We note that such a saddle-point view of debiasing the extra variance term leads to a useful mechanism for better bias-variance trade-offs, leading to the final primal-dual formulation we aim to solve in the next section:

$$\min_{V,\pi\in\mathcal{P}} \max_{\rho \in \mathcal{F}_{\mathcal{S}\times\mathcal{A}}} L_\eta(V,\pi;\rho) := \mathbb{E}_{s,a,s'}\left[\left(\delta(s,a,s') - V(s)\right)^2\right] - \eta\, \mathbb{E}_{s,a,s'}\left[\left(\delta(s,a,s') - \rho(s,a)\right)^2\right], \quad (10)$$

where $\eta \in [0,1]$ is a hyper-parameter controlling the trade-off. When $\eta = 1$, this reduces to the original saddle-point formulation (8). When $\eta = 0$, this reduces to the surrogate objective used in PCL.

4. Smoothed Bellman Error Embedding

In this section, we derive the Smoothed Bellman Error EmbeDding (SBEED) algorithm, based on stochastic mirror descent (Nemirovski et al., 2009), to solve the smoothed Bellman equation. For simplicity of exposition, we mainly discuss the one-step optimization (10), although it is possible to generalize the algorithm to the multi-step and eligibility-traces settings our extened version.

Due to the curse of dimensionality, the quantities (V,π,ρ) are often represented by compact, parametric functions in practice. Denote these parameters by $w = (w_V, w_\pi, w_\rho)$. Abusing notation a little bit, we now write the objective function $L_\eta(V,\pi;\rho)$ as $L_\eta(w_V, w_\pi; w_\rho)$.

First, we note that the inner (dual) problem is standard least-squares regression with parameter w_ρ, so can be solved using a variety of algorithms (Bertsekas, 2016); in the presence of special structures like convexity, global optima can be found efficiently (Boyd & Vandenberghe, 2004). The more involved part is to optimize the primal (w_V, w_π), whose gradients are given by the following theorem.

Theorem 4 (Primal gradient) *Define* $\bar{\ell}_\eta(w_V, w_\pi) := L_\eta(w_V, w_\pi; w_\rho^*)$, *where* $w_\rho^* = \arg\max_{w_\rho} L_\eta(w_V, w_\pi; w_\rho)$. *Let* $\delta_{s,a,s'}$ *be a shorthand for* $\delta(s,a,s')$, *and* $\hat{\rho}$ *be dual parameterized by* w_ρ^*. *Then,*

$$\nabla_{w_V} \bar{\ell}_\eta = 2\mathbb{E}_{s,a,s'}\left[\left(\delta_{s,a,s'} - V(s)\right)\left(\gamma \nabla_{w_V} V(s') - \nabla_{w_V} V(s)\right)\right] - 2\eta\gamma \mathbb{E}_{s,a,s'}\left[\left(\delta_{s,a,s'} - \hat{\rho}(s,a)\right)\nabla_{w_V} V(s')\right],$$

$$\nabla_{w_\pi} \bar{\ell}_\eta = -2\lambda \mathbb{E}_{s,a,s'}\left[(1-\eta)\delta_{s,a,s'} \cdot \nabla_{w_\pi} \log \pi(a|s) + \left(\eta\hat{\rho}(s,a) - V(s)\right) \cdot \nabla_{w_\pi} \log \pi(a|s)\right].$$

Algorithm 1 Online SBEED learning with experience replay

1: Initialize $w = (w_V, w_\pi, w_\rho)$ and π_b randomly, set ϵ.
2: **for** episode $i = 1, \ldots, T$ **do**
3: **for** size $k = 1, \ldots, K$ **do**
4: Add new transition (s,a,r,s') into $\mathcal{D}$ by executing behavior policy π_b.
5: **end for**
6: **for** iteration $j = 1, \ldots, N$ **do**
7: Update w_ρ^j by solving $\min_{w_\rho} \widehat{\mathbb{E}}_{\{s,a,s'\}\sim\mathcal{D}}\left[\left(\delta(s,a,s') - \rho(s,a)\right)^2\right]$.
8: Decay the stepsize ζ_j in rate $\mathcal{O}(1/j)$.
9: Compute the stochastic gradients w.r.t. w_V and w_π as $\widehat{\nabla}_{w_V}\bar{\ell}(V,\pi)$ and $\widehat{\nabla}_{w_\pi}\bar{\ell}(V,\pi)$.
10: Update the parameters of primal function by solving the prox-mappings, *i.e.*,
 update V: $w_V^j = P_{w_V^{j-1}}(\zeta_j \widehat{\nabla}_{w_V}\bar{\ell}(V,\pi))$
 update π: $w_\pi^j = P_{w_\pi^{j-1}}(\zeta_j \widehat{\nabla}_{w_\pi}\bar{\ell}(V,\pi))$
11: **end for**
12: Update behavior policy $\pi_b = \pi^N$.
13: **end for**

With gradients given above, we may apply stochastic mirror descent to update w_V and w_π; that is, given a stochastic gradient direction (for either w_V or w_π), we solve the following prox-mapping in each iteration,

$$P_{z_V}(g) = \operatorname*{argmin}_{w_V} \langle w_V, g\rangle + D_V(w_V, z_V),$$
$$P_{z_\pi}(g) = \operatorname*{argmin}_{w_\pi} \langle w_\pi, g\rangle + D_\pi(w_\pi, z_\pi),$$

where z_V and z_π can be viewed the current weight, and $D_V(w,z)$ and $D_\pi(w,z)$ are Bregman divergences. We can use Euclidean metric for both w_V and w_π, and possibly KL-divergence for w_π. The per-iteration computation complexity is therefore very low, and the algorithm can be scaled up to complex nonlinear approximations.

Algorithm 1 instantiates SBEED, combined with experience replay (Lin, 1992) for greater data efficiency, in an online RL setting. New samples are added to the experience replay buffer $\mathcal{D}$ at the beginning of each episode (Lines 3–5) with a behavior policy. Lines 6–11 correspond to the stochastic mirror descent updates on the primal parameters. Line 12 sets the behavior policy to be the current policy estimate, although other choices may be used. For example, π_b can be a fixed policy (Antos et al., 2008), which is the case we will analyze in the next section.

Remark (Role of dual variables): The dual variable is obtained by solving

$$\min_{\rho} \mathbb{E}_{s,a,s'}\left[\left(R(s,a) + \gamma V(s') - \lambda \log \pi(a|s) - \rho(s,a)\right)^2\right].$$

The solution to this optimization problem is

$$\rho^*(s,a) = R(s,a) + \gamma \mathbb{E}_{s'|s,a}\left[V(s')\right] - \lambda \log \pi(a|s).$$

Therefore, the dual variables try to approximate the one-step smoothed Bellman backup values, given a (V, π) pair. Similarly, in the equivalent (8), the optimal dual variable $\nu(s,a)$ is to fit the one-step smoothed Bellman error. Therefore, each iteration of SBEED could be understood as first fitting a parametric model to the one-step Bellman backups (or equivalently, the one-step Bellman error), and then applying stochastic mirror descent to adjust V and π.

Remark (Connection to TRPO and NPG): The update of w_π is related to trust region policy optimization (TRPO) (Schulman et al., 2015) and natural policy gradient (NPG) (Kakade, 2002; Rajeswaran et al., 2017) when D_π is the KL-divergence. Specifically, in Kakade (2002) and Rajeswaran et al. (2017), w_π is updated by $\operatorname{argmin}_{w_\pi} \mathbb{E}\left[\langle w_\pi, \nabla_{w_\pi} \log \pi^t(a|s) A(a,s)\rangle\right] + \frac{1}{\eta}\mathrm{KL}(\pi_{w_\pi}||\pi_{w_\pi^{old}})$, which is similar to $P_{w_\pi^{j-1}}$ with the difference in replacing the $\log \pi^t(a|s)A(a,s)$ with our gradient. In Schulman et al. (2015), a related optimization with hard constraints is used for policy updates: $\min_{w_\pi} \mathbb{E}\left[\pi(a|s)A(a,s)\right]$, such that $\mathrm{KL}(\pi_{w_\pi}||\pi_{w_\pi^{old}}) \leqslant \eta$. Although these operations are similar to $P_{w_\pi^{j-1}}$, we emphasize that the estimation of the advantage function, $A(s,a)$, and the update of policy are separated in NPG and TRPO. Arbitrary policy evaluation algorithm can be adopted for estimating the value function for *current* policy. While in our algorithm, $(1-\eta)\delta(s,a) + \eta\rho^*(s,a) - V(s)$ is different from the vanilla advantage function, which is designed for off-policy learning particularly, and the estimation of $\rho(s,a)$ and $V(s)$ is also integrated as the whole part.

5. Theoretical Analysis

In this section, we give a theoretical analysis for our algorithm in the same setting of Antos et al. (2008) where samples are prefixed and from *one single β-mixing off-policy sample path*. For simplicity, we consider the case that applying the algorithm for $\eta = 1$ with the equivalent optimization (8). The analysis is applicable to (9) directly. There are three groups of results. First, in Section 5.1, we show that under appropriate choices of stepsize and prox-mapping, SBEED converges to a stationary point of the finite-sample approximation (i.e., empirical risk) of the optimization (8). Second, in Section 5.2, we analyze generalization error of SBEED. Finally, in Section 5.3, we give an overall performance bound for the algorithm, by combining four sources of errors: (i) optimization error, (ii) generalization error, (iii) bias induced by Nesterov smoothing, and (iv) approximation error induced by using function approximation.

Notations. Denote by $\mathcal{V}_w$, $\mathcal{P}_w$ and $\mathcal{H}_w$ the parametric function classes of value function V, policy π, and dual variable ν, respectively. Denote the total number of steps in the given off-policy trajectory as T. We summarize the notations for the objectives after parametrization and finite-sample approximation and their corresponding optimal solutions in the table for reference:

	minimax obj.	primal obj.	optimum
original	$L(V,\pi;\nu)$	$\ell(V,\pi)$	$(V_\lambda^*, \pi_\lambda^*)$
parametric	$L_w(V_w,\pi_w;\nu_w)$	$\ell_w(V_w,\pi_w)$	(V_w^*, π_w^*)
empirical	$\widehat{L}_T(V_w,\pi_w;\nu_w)$	$\widehat{\ell}_T(V_w,\pi_w)$	$(\widehat{V}_w^*, \widehat{\pi}_w^*)$

Denote the L_2 norm of a function f w.r.t. $\mu(s)\pi_b(a|s)$ by $\|f\|^2 := \int f(s,a)^2 \mu(s)\pi_b(a|s)dsda$. We introduce a scaled norm :

$$\|V\|^2_{\mu\pi_b} := \int \left(\gamma \mathbb{E}_{s'|s,a}\left[V(s')\right] - V(s)\right)^2 \mu(s)\pi_b(a|s)dsda$$

for value function; this is indeed a well-defined norm since $\|V\|^2_{\mu\pi_b} = \|(\gamma P - I)V\|_2^2$ and $I - \gamma P$ is injective.

5.1. Convergence Analysis

It is well-known that for convex-concave saddle-point problems, applying stochastic mirror descent ensures global convergence in a sublinear rate (Nemirovski et al., 2009). However, this result no longer holds for problems without convex-concavity. Our SBEED algorithm, on the other hand, can be regarded as a special case of the stochastic mirror descent algorithm for solving the non-convex primal minimization problem $\min_{V_w,\pi_w} \widehat{\ell}_T(V_w,\pi_w)$. The latter was proven to converge sublinearly to a stationary point when stepsize is diminishing and Euclidean distance is used for the prox-mapping (Ghadimi & Lan, 2013). For completeness, we list the result below.

Theorem 5 (Convergence, Ghadimi & Lan (2013)) *Consider the case when Euclidean distance is used in the algorithm. Assume that the parametrized objective $\widehat{\ell}_T(V_w,\pi_w)$ is K-Lipschitz and variance of its stochastic gradient is bounded by σ^2. Let the algorithm run for N iterations with stepsize $\zeta_k = \min\{\frac{1}{K}, \frac{D'}{\sigma\sqrt{N}}\}$ for some $D' > 0$ and output $w^1, \ldots, w^N$. Setting the candidate solution to be $(\widehat{V}_w^N, \widehat{\pi}_w^N)$ with w randomly chosen from $w^1, \ldots, w^N$ such that $P(w = w^j) = \frac{2\zeta_j - K\zeta_j^2}{\sum_{j=1}^N (2\zeta_j - K\zeta_j^2)}$, then it holds that $\mathbb{E}\left[\left\|\nabla \widehat{\ell}_T(\widehat{V}_w^N, \widehat{\pi}_w^N)\right\|^2\right] \leqslant \frac{KD^2}{N} + (D' + \frac{D}{D'})\frac{\sigma}{\sqrt{N}}$ where $D := \sqrt{2(\widehat{\ell}_T(V_w^1,\pi_w^1) - \min \widehat{\ell}_T(V_w,\pi_w))/K}$ represents the distance of the initial solution to the optimal solution.*

The above result implies that the algorithm converges sublinearly to a stationary point, whose rate will depend on the smoothing parameter.

In practice, once we parametrize the dual function, ν or ρ, with neural networks, we cannot achieve the optimal parameters. However, we can still achieve convergence by applying the stochastic gradient descent to a (statistical) local Nash equilibrium asymptotically. We provided the variant of SBEED algorithm and the convergence analysis in our extended version.

5.2. Statistical Error

In this section, we characterize the statistical error, namely, $\epsilon_{\text{stat}}(T) := \ell_w(\widehat{V}_w^*, \widehat{\pi}_w^*) - \ell_w(V_w^*, \pi_w^*)$, induced by learning with finite samples. We first make the following standard assumptions about the MDPs:

Assumption 1 (MDP regularity) *Assume* $\|R(s,a)\|_\infty \leqslant C_R$ *and that there exists an optimal policy,* $\pi_\lambda^*(a|s)$*, such that* $\|\log \pi_\lambda^*(a|s)\|_\infty \leqslant C_\pi$.

Assumption 2 (Sample path property, Antos et al. (2008)) *Denote by* $\mu(s)$ *the stationary distribution of behavior policy* π_b *over the MDP. We assume* $\pi_b(a|s) > 0$, $\forall\,(s,a) \in \mathcal{S}\times\mathcal{A}$*, and the corresponding Markov process* $P^{\pi_b}(s'|s)$ *is ergodic. We further assume that* $\{s_i\}_{i=1}^T$ *is strictly stationary and exponentially* β*-mixing with a rate defined by the parameters* (b,κ)[2].

Assumption 1 ensures the solvability of the MDP and boundedness of the optimal value functions, V^* and V_λ^*. Assumption 2 ensures the β-mixing property of the samples $\{(s_i,a_i,R_i)\}_{i=1}^T$ (see, *e.g.*, Proposition 4 in Carrasco & Chen (2002)), which is often necessary to obtain large deviation bounds.

Invoking a generalized version of Pollard's tail inequality to β-mixing sequences and prior results in Antos et al. (2008) and Haussler (1995), we show that

Theorem 6 (Statistical error) *Under Assumption 2, it holds with at least probability* $1-\delta$ *that*

$$\epsilon_{stat}(T) \leqslant 2\sqrt{\frac{M\left(\max\left(M/b,1\right)\right)^{1/\kappa}}{C_2 T}},$$

where M, C_2 *are some constants.*

5.3. Error Decomposition

As one shall see, the error between $(\widehat{V}_w^N, \widehat{w}^N)$ (optimal solution to the finite sample problem) and the true solution (V^*,π^*) to the Bellman equation consists of three parts: (i) the error introduced by smoothing, which has been characterized in Section 3.1, (ii) the approximation error, which is tied to the flexibility of the parametrized function classes $\mathcal{V}_w$, $\mathcal{P}_w$, $\mathcal{H}_w$, and (iii) the statistical error. More specifically, we arrive at the following explicit decomposition:

Specifically, we arrive at the following explicit decomposition, where $\epsilon_{\text{app}}^\pi := \sup_{\pi\in\mathcal{P}}\inf_{\pi'\in\mathcal{P}_w}\|\pi-\pi'\|_\infty$ is the function approximation error between $\mathcal{P}_w$ and $\mathcal{P}$, and ϵ_{app}^V and $\epsilon_{\text{app}}^\nu$ are the approximation errors for V and ν, respectively.

Theorem 7 *Under Assumptions 1 and 2, it holds that* $\left\|\widehat{V}_w^N - V^*\right\|_{\mu\pi_b}^2 \leqslant 12(K+C_\infty)\epsilon_{app}^\nu + 2C_\nu(1+\gamma)\epsilon_{app}^V(\lambda) + 6C_\nu\epsilon_{app}^\pi(\lambda) + 16\lambda^2C_\pi^2 + \left(2\gamma^2+2\right)\left(\frac{\gamma\lambda}{1-\gamma}H^*\right)^2 + 2\epsilon_{stat}(T) + 2\left\|\widehat{V}_w^N - \widehat{V}_w^*\right\|_{\mu\pi_b}^2$, *where* $C_\infty := \max\left\{\frac{C_R}{1-\gamma}, C_\pi\right\}$ *and* $C_\nu := \max_{\nu\in\mathcal{H}_w}\|\nu\|_2$.

Detailed proof can be found in the extended version. Ignoring the constant factors, the above results can be simplified as

$$\left\|\widehat{V}_w^N - V^*\right\|_{\mu\pi_b}^2 \leqslant \epsilon_{\text{app}}(\lambda) + \epsilon_{\text{sm}}(\lambda) + \epsilon_{\text{stat}}(T) + \epsilon_{\text{opt}},$$

where $\epsilon_{\text{app}}(\lambda) := \mathcal{O}(\epsilon_{\text{app}}^\nu + \epsilon_{\text{app}}^V(\lambda) + \epsilon_{\text{app}}^\pi(\lambda))$ corresponds to the approximation error, $\epsilon_{\text{sm}}(\lambda) := \mathcal{O}(\lambda^2)$ corresponds to the bias induced by smoothing, and $\epsilon_{\text{stat}}(T) := \mathcal{O}(1/\sqrt{T})$ corresponds to the statistical error.

There exists a delicate trade-off between the smoothing bias and approximation error. Using large λ increases the smoothing bias but decreases the approximation error since the solution function space is better behaved. The concrete correspondence between λ and $\epsilon_{\text{app}}(\lambda)$ depends on the specific form of the function approximators, which is beyond the scope of this paper. Finally, when the approximation is good enough (i.e., zero approximation error and full column rank of feature matrices), then our algorithm will converge to the optimal value function V^* as $\lambda\to 0$ and $(N,T)\to\infty$.

6. Related Work

One of our main contributions is a provably convergent algorithm when nonlinear approximation is used in the off-policy control case. Convergence guarantees exist in the literature for a few rather special cases, as reviewed in the introduction (Boyan & Moore, 1995; Gordon, 1995; Tsitsiklis & Van Roy, 1997; Ormoneit & Sen, 2002; Antos et al., 2008; Melo et al., 2008). Of particular interest is the Greedy-GQ algorithm (Maei et al., 2010), who uses two time-scale analysis to shown asymptotic convergence only for *linear* function approximation in the controlled case. However, it does not take the true gradient estimator in the algorithm, and the update of policy may become intractable when the action space is continuous.

Algorithmically, our method is most related to RL algorithms with entropy-regularized policies. Different from the motivation in our method where the entropy regularization is introduced in the dual form for smoothing (Nesterov, 2005), the entropy-regularized MDP has been proposed for exploration (de Farias & Van Roy, 2000; Haarnoja et al., 2017), taming noise in observations (Rubin et al., 2012; Fox et al., 2016), and ensuring tractability (Todorov, 2006). Specifically, Fox et al. (2016) proposed soft Q-learning for the tabular case, but its extension to the function approximation case is hard, as the summation operation in log-sum-exp of the update rule becomes a computationally expensive integration. To avoid such a difficulty, Haarnoja et al. (2017) approximate the integral by Monte Carlo using the Stein variational gradient descent sampler, but limited

[2] A β-mixing process is said to mix at an exponential rate with parameter $b,\kappa>0$ if $\beta_m = O(\exp(-bm^{-\kappa}))$.

theory is provided. Another related algorithm is developed by Asadi & Littman (2017) for the tabular case, which resembles SARSA with a particular policy; also see Liu et al. (2017) for a Bayesian variant. Observing the duality connection between soft Q-learning and maximum entropy policy optimization, Neu et al. (2017) and Schulman et al. (2017) investigate the equivalence between these two types of algorithms.

Besides the difficulty to generalize these algorithms to multi-step trajectories in off-policy setting, the major drawback of these algorithms is the lack of theoretical guarantees when combined with function approximation. It is not clear whether the algorithms converge or not, let alone the quality of the stationary points. That said, Nachum et al. (2017; 2018) also exploit the consistency condition in Theorem 3 and propose the PCL algorithm which optimizes the upper bound of the mean squared consistency Bellman error (7). The same consistency condition is also discovered in Rawlik et al. (2012), and the proposed Φ-learning algorithm can be viewed as a fix-point iteration version of the PCL with a tabular Q-function. However, as we discussed in Section 3, the PCL algorithms becomes biased in stochastic environment, which may lead to inferior solutions Baird (1995).

Several recent works (Chen & Wang, 2016; Wang, 2017; Dai et al., 2018) have also considered saddle-point formulations of Bellman equations, but these formulations are fundamentally different from ours. These saddle-point problems are derived from the *Lagrangian* dual of the linear programming formulation of Bellman equations (Schweitzer & Seidmann, 1985; de Farias & Van Roy, 2003). In contrast, our formulation is derived from the Bellman equation directly using *Fenchel* duality/transformation. It would be interesting to investigate the connection between these two saddle-point formulations in future work.

7. Experiments

The goal of our experimental evalution is two folds: (i) to better understand of the effect of each algorithmic component in the proposed algorithm; (ii) to demonstrate the stability and efficiency of SBEED in both *off-policy* and *on-policy* settings. Therefore, we conducted an ablation study on SBEED, and a comprehensive comparison to state-of-the-art reinforcement learning algorithms. While we derive and present SBEED for the single-step Bellman error case, it can be extended to multi-step cases as shown in the long version. In our experiment, we used this multi-step version.

7.1. Ablation Study

To get a better understanding of the trade-off between the variance and bias, including both the bias from the smoothing technique and the introduction of the function approximator, we performed ablation study in the Swimmer-v1 environment with *stochastic* transition by varying the coefficient for entropic regularization λ and the coefficient of the dual function η in the optimization (10), as well as the number of the rollout steps, k.

The effect of smoothing. We used entropy regularization to avoid non-smoothness in the squared Bellman error objective, at the cost of an introduced bias. We varied λ and evaluated the performance of SBEED. The results in Figure 1(a) are as expected: there is indeed an intermediate value for λ that gives the best bias/smoothness trade-off.

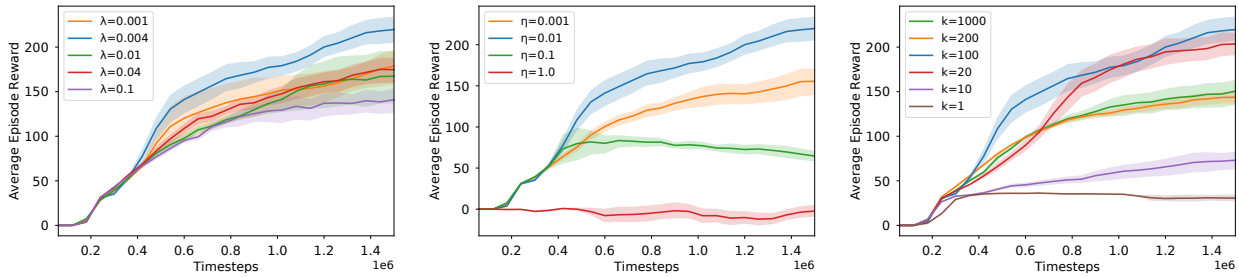
The effect of dual function. One of the important components in our algorithm is the dual function, which cancels the variance. The effect of such cancellation is controlled by $\eta \in [0, 1]$, and we expected an intermediate value gives the best performance. This is verified by the experiment of varying η, as shown in Figure 1(b).

The effect of multi-step. SBEED can be extended to the multi-step version. However, increasing the length of lookahead will also increase the variance. We tested the performance of the algorithm with different lookahead lengths (denoted by k). The results shown in Figure 1(c) confirms that an intermediate value for k yields the best result.

7.2. Comparison in Continuous Control Tasks

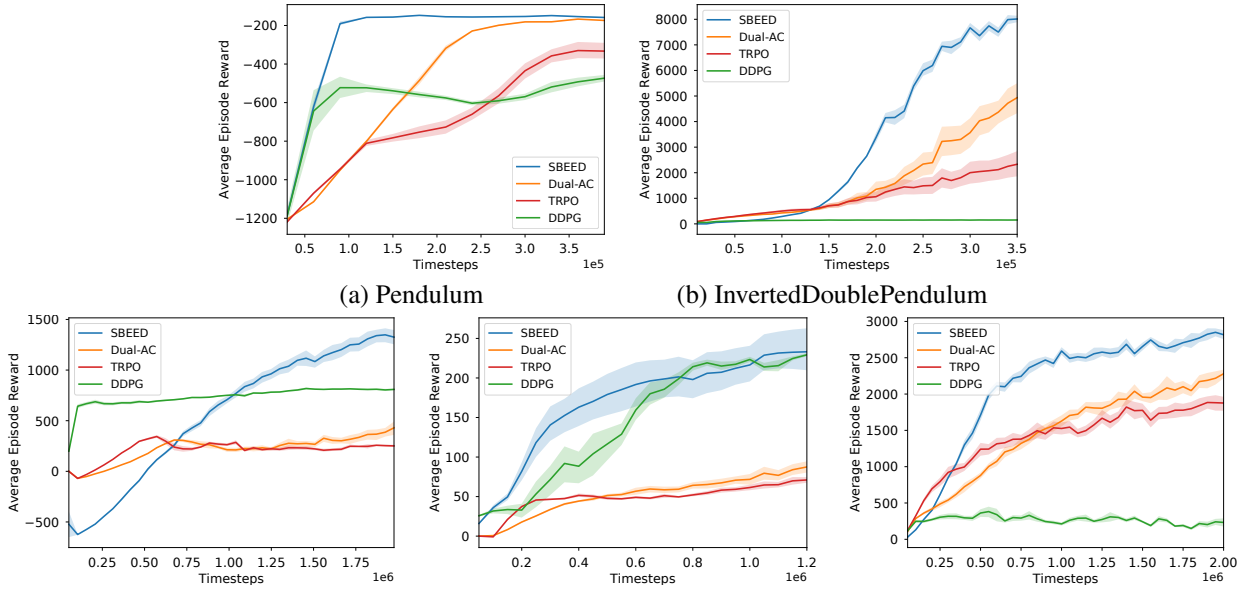
We tested SBEED across multiple continuous control tasks from the OpenAI Gym benchmark (Brockman et al., 2016) using the MuJoCo simulator (Todorov et al., 2012), including Pendulum-v0, InvertedDoublePendulum-v1, HalfCheetah-v1, Swimmer-v1, and Hopper-v1. For fairness, we follows the default setting of the MuJoCo simulator in each task in this section. These tasks have dynamics of different natures, so are helpful for evaluating the behavior of the proposed SBEED in different scenarios. We compared SBEED with several state-of-the-art algorithms, including two on-policy algorithms, trust region policy optimization (TRPO) (Schulman et al., 2015) dual actor-critic (Dual AC) (Dai et al., 2018), and one off-policy algorithm, deep deterministic policy gradient (DDPG) (Lillicrap et al., 2015). We did not include PCL (Nachum et al., 2017) as it is a special case of our algorithm by setting $\eta = 0$, *i.e.*, ignoring the updates for dual function. Since TRPO and Dual-AC are only applicable for the on-policy setting, for fairness, we also conducted the comparison with these two algorithm in on-policy setting. Due to the space limitation, these results are provided in the extened version.

We ran the algorithms with 5 random seeds and reported the average rewards with 50% confidence intervals. The results are shown in Figure 2. We can see that our SBEED achieves significantly better performance than all other algorithms across the board. These results suggest that the SBEED can exploit the off-policy samples efficiently and stably, and achieve a good trade-off between bias and variance.



(a) λ for entropy (b) η for dual (c) k for steps

Figure 1. Ablation study of the SBEED on Swimmer-v1. We vary λ, η, and k to justify three major components in our algorithm.



(a) Pendulum (b) InvertedDoublePendulum

(c) HalfCheetah (d) Swimmer (e) Hopper

Figure 2. The results of SBEED against TRPO, Dual AC and DDPG. Each plot shows the average reward during training across 5 random runs, with 50% confidence interval. The x-axis is the number of training iterations. SBEED achieves significantly better performance than the competitors on all tasks.

It should be emphasized that the stability of algorithm is an important issue in reinforcement learning. As we can see from the results, although DDPG can also exploit the off-policy sample, which promotes its efficiency in stable environments, *e.g.*, HalfCheetah-v1 and Swimmer-v1, it may fail to learn in unstable environments, *e.g.*, InvertedDoublePendulum-v1 and Hopper-v1, which was observed by Henderson et al. (2018) and Haarnoja et al. (2018). In contrast, SBEED is consistently reliable and effective in different tasks.

8. Conclusion

We provided a new optimization perspective of the Bellman equation, based on which we developed the new SBEED algorithm for policy optimization in reinforcement learning. The algorithm is *provably convergent* even when *nonlinear* function approximation is used on *off-policy* samples. We also provided a PAC bound for its sample complexity based on *one single off-policy sample path* collected by a fixed behavior policy. Empirical study shows the proposed algorithm achieves superior performance across the board, compared to state-of-the-art baselines on several MuJoCo control tasks.

Acknowledgments

Part of this work was done during BD's internship at Microsoft Research, Redmond. Part of the work was done when LL and JC were with Microsoft Research, Redmond. We thank Mohammad Ghavamzadeh, Nan Jiang, Csaba Szepesvari, and Greg Tucker for their insightful comments and discussions. NH is supported by NSF CCF-1755829. LS is supported in part by NSF IIS-1218749, NIH BIGDATA 1R01GM108341, NSF CAREER IIS-1350983, NSF IIS-1639792 EAGER, NSF CNS-1704701, ONR N00014-15-1-2340, Intel ISTC, NVIDIA and Amazon AWS.

References

Antos, András, Szepesvári, Csaba, and Munos, Rémi. Learning near-optimal policies with Bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning*, 71(1):89–129, 2008.

Asadi, Kavosh and Littman, Michael L. An alternative softmax operator for reinforcement learning. In *ICML*, pp. 243–252, 2017.

Baird, Leemon. Residual algorithms: Reinforcement learning with function approximation. In *ICML*, pp. 30–37. Morgan Kaufmann, 1995.

Bertsekas, Dimitri P. *Nonlinear Programming*. Athena Scientific, 3rd edition, 2016. ISBN 978-1-886529-05-2.

Bertsekas, Dimitri P. and Tsitsiklis, John N. *Neuro-Dynamic Programming*. Athena Scientific, September 1996. ISBN 1-886529-10-8.

Boyan, Justin A. Least-squares temporal difference learning. *Machine Learning*, 49(2):233–246, November 2002.

Boyan, Justin A. and Moore, Andrew W. Generalization in reinforcement learning: Safely approximating the value function. In *NIPS*, pp. 369–376, 1995.

Boyd, Stephen and Vandenberghe, Lieven. *Convex Optimization*. Cambridge University Press, Cambridge, England, 2004.

Brockman, Greg, Cheung, Vicki, Pettersson, Ludwig, Schneider, Jonas, Schulman, John, Tang, Jie, and Zaremba, Wojciech. OpenAI Gym, 2016. arXiv:1606.01540.

Carrasco, Marine and Chen, Xiaohong. Mixing and moment properties of various GARCH and stochastic volatility models. *Econometric Theory*, 18(1):17–39, 2002.

Chen, Yichen and Wang, Mengdi. Stochastic primal-dual methods and sample complexity of reinforcement learning. *arXiv preprint arXiv:1612.02516*, 2016.

Dai, Bo, He, Niao, Pan, Yunpeng, Boots, Byron, and Song, Le. Learning from conditional distributions via dual embeddings. In *AISTATS*, pp. 1458–1467, 2017.

Dai, Bo, Shaw, Albert, He, Niao, Li, Lihong, and Song, Le. Boosting the actor with dual critic. *ICLR*, 2018. arXiv:1712.10282.

de Farias, Daniela Pucci and Van Roy, Benjamin. On the existence of fixed points for approximate value iteration and temporal-difference learning. *Journal of Optimization Theory and Applications*, 105(3):589–608, 2000.

de Farias, Daniela Pucci and Van Roy, Benjamin. The linear programming approach to approximate dynamic programming. *Operations Research*, 51(6):850–865, 2003.

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

Fox, Roy, Pakman, Ari, and Tishby, Naftali. Taming the noise in reinforcement learning via soft updates. In *UAI*, 2016.

Ghadimi, Saeed and Lan, Guanghui. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

Gordon, Geoffrey J. Stable function approximation in dynamic programming. In *ICML*, pp. 261–268, 1995.

Haarnoja, Tuomas, Tang, Haoran, Abbeel, Pieter, and Levine, Sergey. Reinforcement learning with deep energy-based policies. In *ICML*, pp. 1352–1361, 2017.

Haarnoja, Tuomas, Zhou, Aurick, Abbeel, Pieter, and Levine, Sergey. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

Haussler, David. Sphere packing numbers for subsets of the Boolean n-cube with bounded Vapnik-Chervonenkis dimension. *Journal of Combinatorial Theory, Series A*, 69(2):217–232, 1995.

Henderson, Peter, Islam, Riashat, Bachman, Philip, Pineau, Joelle, Precup, Doina, and Meger, David. Deep reinforcement learning that matters. In *AAAI*, 2018.

Kakade, Sham. A natural policy gradient. In *NIPS*, pp. 1531–1538, 2002.

Lagoudakis, Michail G. and Parr, Ronald. Least-squares policy iteration. *JMLR*, 4:1107–1149, 2003.

Lillicrap, Timothy P, Hunt, Jonathan J, Pritzel, Alexander, Heess, Nicolas, Erez, Tom, Tassa, Yuval, Silver, David, and Wierstra, Daan. Continuous control with deep reinforcement learning. *arXiv:1509.02971*, 2015.

Lin, Long-Ji. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning*, 8 (3–4):293–321, 1992.

Liu, Bo, Liu, Ji, Ghavamzadeh, Mohammad, Mahadevan, Sridhar, and Petrik, Marek. Finite-sample analysis of proximal gradient td algorithms. In *UAI*, 2015.

Liu, Yang, Ramachandran, Prajit, Liu, Qiang, and Peng, Jian. Stein variational policy gradient. In *UAI*, 2017.

Macua, Sergio Valcarcel, Chen, Jianshu, Zazo, Santiago, and Sayed, Ali H. Distributed policy evaluation under multiple behavior strategies. *IEEE Transactions on Automatic Control*, 60(5):1260–1274, 2015.

Maei, Hamid Reza. *Gradient Temporal-Difference Learning Algorithms*. PhD thesis, University of Alberta, Edmonton, Alberta, Canada, 2011.

Maei, Hamid Reza, Szepesvári, Csaba, Bhatnagar, Shalabh, and Sutton, Richard S. Toward off-policy learning control with function approximation. In *ICML*, pp. 719–726, 2010.

Mahadevan, Sridhar, Liu, Bo, Thomas, Philip S., Dabney, William, Giguere, Stephen, Jacek, Nicholas, Gemp, Ian, and Liu, Ji. Proximal reinforcement learning: A new theory of sequential decision making in primal-dual spaces. CoRR abs/1405.6757, 2014.

Melo, Francisco S., Meyn, Sean P., and Ribeiro, M. Isabel. An analysis of reinforcement learning with function approximation. In *ICML*, pp. 664–671, 2008.

Mnih, Volodymyr, Kavukcuoglu, Koray, Silver, David, Rusu, Andrei A., Veness, Joel, Bellemare, Marc G., Graves, Alex, Riedmiller, Martin, Fidjeland, Andreas K., Ostrovski, Georg, Petersen, Stig, Beattie, Charles, Sadik, Amir, Antonoglou, Ioannis, King, Helen, Kumaran, Dharshan, Wierstra, Daan, Legg, Shane, and Hassabis, Demis. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015.

Mnih, Volodymyr, Badia, Adrià Puigdomènech, Mirza, Mehdi, Graves, Alex, Lillicrap, Timothy P., Harley, Tim, Silver, David, and Kavukcuoglu, Koray. Asynchronous methods for deep reinforcement learning. In *ICML*, pp. 1928–1937, 2016.

Nachum, Ofir, Norouzi, Mohammad, Xu, Kelvin, and Schuurmans, Dale. Bridging the gap between value and policy based reinforcement learning. In *NIPS*, pp. 2772–2782, 2017.

Nachum, Ofir, Norouzi, Mohammad, Xu, Kelvin, and Schuurmans, Dale. Trust-PCL: An off-policy trust region method for continuous control. In *ICLR*, 2018. arXiv:1707.01891.

Nemirovski, Arkadi, Juditsky, Anatoli, Lan, Guanghui, and Shapiro, Alexander. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 19 (4):1574–1609, 2009.

Nesterov, Yu. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1):127–152, 2005.

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

Ormoneit, Dirk and Sen, Śaunak. Kernel-based reinforcement learning. *Machine Learning*, 49:161–178, 2002.

Puterman, Martin L. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.

Rajeswaran, Aravind, Lowrey, Kendall, Todorov, Emanuel V., and Kakade, Sham M. Towards generalization and simplicity in continuous control. In *NIPS*, 2017.

Rawlik, Konrad, Toussaint, Marc, and Vijayakumar, Sethu. On stochastic optimal control and reinforcement learning by approximate inference. In *Robotics: Science and Systems VIII*, 2012.

Rubin, Jonathan, Shamir, Ohad, and Tishby, Naftali. Trading value and information in MDPs. *Decision Making with Imperfect Decision Makers*, pp. 57–74, 2012.

Rummery, G. A. and Niranjan, M. On-line Q-learning using connectionist systems. Technical Report CUED/F-INFENG/TR 166, Cambridge University Engineering Department, 1994.

Schulman, John, Levine, Sergey, Abbeel, Pieter, Jordan, Michael I, and Moritz, Philipp. Trust region policy optimization. In *ICML*, pp. 1889–1897, 2015.

Schulman, John, Abbeel, Pieter, and Chen, Xi. Equivalence between policy gradients and soft Q-learning, 2017. arXiv:1704.06440.

Schweitzer, Paul J. and Seidmann, Abraham. Generalized polynomial approximations in Markovian decision processes. *Journal of Mathematical Analysis and Applications*, 110(2):568–582, 1985.

Shapiro, Alexander, Dentcheva, Darinka, and Ruszczyński, Andrzej. *Lectures on Stochastic Programming: Modeling and Theory*. SIAM, 2009. ISBN 978-0-89871-687-0.

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

Sutton, Richard S. Generalization in reinforcement learning: Successful examples using sparse coarse coding. In *NIPS*, pp. 1038–1044, 1996.

Sutton, Richard S and Barto, Andrew G. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

Sutton, Richard S., Maei, Hamid Reza, Precup, Doina, Bhatnagar, Shalabh, Silver, David, Szepesvri, Csaba, and Wiewiora, Eric. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *ICML*, pp. 993–1000, 2009.

Todorov, Emanuel. Linearly-solvable Markov decision problems. In *NIPS*, pp. 1369–1376, 2006.

Todorov, Emanuel, Erez, Tom, and Tassa, Yuval. Mujoco: A physics engine for model-based control. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 5026–5033. IEEE, 2012.

Tsitsiklis, John N. and Van Roy, Benjamin. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 42:674–690, 1997.

Wang, Mengdi. Randomized linear programming solves the discounted Markov decision problem in nearly-linear running time. *ArXiv e-prints*, 2017.

Watkins, Christopher J.C.H. *Learning from Delayed Rewards*. PhD thesis, King's College, University of Cambridge, UK, 1989.

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