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* (*SBEED*), 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 the long-term return 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.

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 is 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 as a *shaping* reward added to the reward function of an induced, equivalent MDP; see Appendix C.2 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's 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 algorithm which uses 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. Proofs are deferred to Appendix A.

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 following important 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 an *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 (Appendix C). 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] Appendices are available in the long version of the paper 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 (see Proposition 8 in Appendix B). 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 (Appendices C.2 and C.3).

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 the following 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 $\|V\|^2_{\mu\pi_b} = \|(I-\gamma P)V\|_2^2$ 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 detailed Algorithm 2 and the convergence analysis in Appendix D.3.

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 *and* C_2 *are appropriate constants.*

Detailed proof can be found in Appendix D.2.

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.

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 $\epsilon_{\text{app}}^\nu$ 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 Appendix D.1. 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), 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 theory is provided. Another related algorithm is developed by Asadi

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

& 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 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 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 (Appendix C.2). 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).

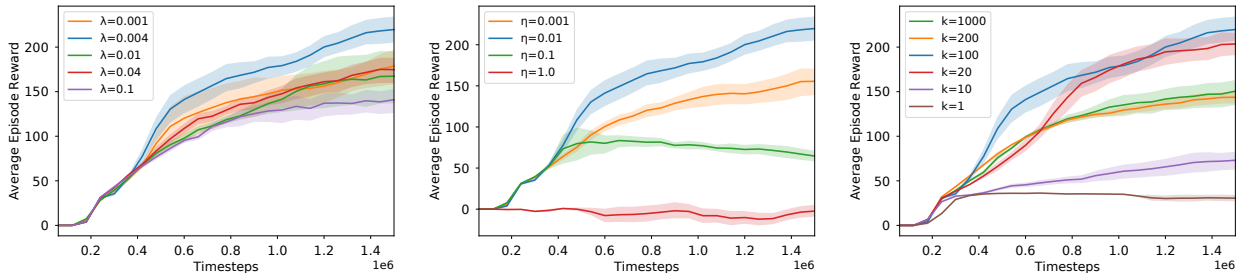
The effect of multi-step. SBEED can be extended to the multi-step version. However, increasing the length of lookahead will also increase the variance. We tested the performance of the algorithm with different lookahead lengths (denoted by k). The results shown in Figure 1(c) confirms that an intermediate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

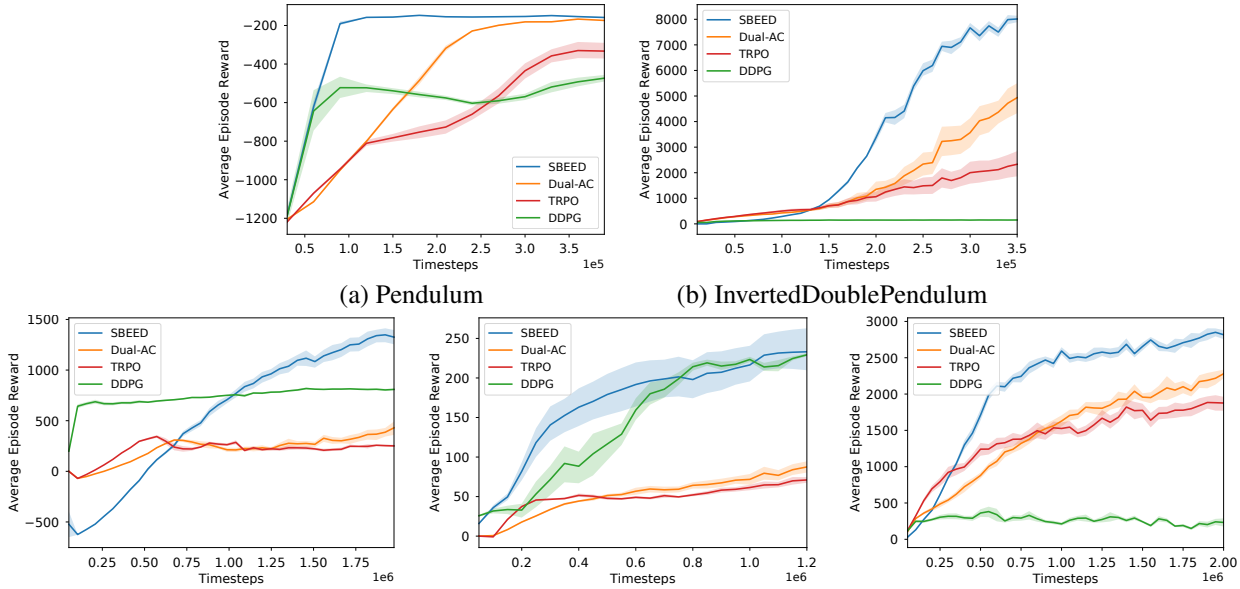
We ran the algorithm with 5 random seeds and reported the average rewards with 50% confidence intervals. The results are shown in Figure 2. We can see that our SBEED achieves significantly better performance than all other algorithms across the board. These results suggest that the SBEED can exploit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

It should be emphasized that the stability of algorithm



(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.

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, A., Szepesvári, C., and Munos, R. 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, K. and Littman, M. L. An alternative softmax operator for reinforcement learning. In *ICML*, pp. 243–252, 2017.

Bach, F. Breaking the curse of dimensionality with convex neural networks. *JMLR*, 18(19):1–53, 2014.

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

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

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

Borkar, V. S. Stochastic approximation with two time scales. *Systems & Control Letters*, 29(5):291–294, 1997.

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

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

Boyd, S. and Vandenberghe, L. *Convex Optimization*. Cambridge University Press, Cambridge, England, 2004.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. OpenAI Gym, 2016. arXiv:1606.01540.

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

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

Dai, B., Xie, B., He, N., Liang, Y., Raj, A., Balcan, M.-F. F., and Song, L. Scalable kernel methods via doubly stochastic gradients. In *NIPS*, pp. 3041–3049, 2014.

Dai, B., He, N., Pan, Y., Boots, B., and Song, L. Learning from conditional distributions via dual embeddings. In *AISTATS*, pp. 1458–1467, 2017.

Dai, B., Shaw, A., He, N., Li, L., and Song, L. Boosting the actor with dual critic. *ICLR*, 2018. arXiv:1712.10282.

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

Du, S. S., Chen, J., Li, L., Xiao, L., and Zhou, D. Stochastic variance reduction methods for policy evaluation. In *ICML*, pp. 1049–1058, 2017.

Fox, R., Pakman, A., and Tishby, N. Taming the noise in reinforcement learning via soft updates. In *UAI*, 2016.

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

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

Haarnoja, T., Tang, H., Abbeel, P., and Levine, S. Reinforcement learning with deep energy-based policies. In *ICML*, pp. 1352–1361, 2017.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

Haussler, D. 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, P., Islam, R., Bachman, P., Pineau, J., Precup, D., and Meger, D. Deep reinforcement learning that matters. In *AAAI*, 2018.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Klambauer, G., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a nash equilibrium. *arXiv preprint arXiv:1706.08500*, 2017.

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

Kearns, M. J. and Singh, S. P. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49(2–3):209–232, 2002.

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

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *arXiv:1509.02971*, 2015.

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

Liu, B., Liu, J., Ghavamzadeh, M., Mahadevan, S., and Petrik, M. Finite-sample analysis of proximal gradient td algorithms. In *UAI*, 2015.

Liu, Y., Ramachandran, P., Liu, Q., and Peng, J. Stein variational policy gradient. In *UAI*, 2017.

Macua, S. V., Chen, J., Zazo, S., and Sayed, A. H. Distributed policy evaluation under multiple behavior strategies. *IEEE Transactions on Automatic Control*, 60(5):1260–1274, 2015.

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

Maei, H. R., Szepesvári, C., Bhatnagar, S., and Sutton, R. S. Toward off-policy learning control with function approximation. In *ICML*, pp. 719–726, 2010.

Mahadevan, S., Liu, B., Thomas, P. S., Dabney, W., Giguere, S., Jacek, N., Gemp, I., and Liu, J. Proximal reinforcement learning: A new theory of sequential decision making in primal-dual spaces. CoRR abs/1405.6757, 2014.

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

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D., and Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In *ICML*, pp. 1928–1937, 2016.

Nachum, O., Norouzi, M., Xu, K., and Schuurmans, D. Bridging the gap between value and policy based reinforcement learning. In *NIPS*, pp. 2772–2782, 2017.

Nachum, O., Norouzi, M., Xu, K., and Schuurmans, D. Trust-PCL: An off-policy trust region method for continuous control. In *ICLR*, 2018. arXiv:1707.01891.

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

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

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

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

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

Rajeswaran, A., Lowrey, K., Todorov, E. V., and Kakade, S. M. Towards generalization and simplicity in continuous control. In *NIPS*, 2017.

Rawlik, K., Toussaint, M., and Vijayakumar, S. On stochastic optimal control and reinforcement learning by approximate inference. In *Robotics: Science and Systems VIII*, 2012.

Rubin, J., Shamir, O., and Tishby, N. 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, J., Levine, S., Abbeel, P., Jordan, M. I., and Moritz, P. Trust region policy optimization. In *ICML*, pp. 1889–1897, 2015.

Schulman, J., Abbeel, P., and Chen, X. Equivalence between policy gradients and soft Q-learning, 2017. arXiv:1704.06440.

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

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

Strehl, A. L., Li, L., and Littman, M. L. Reinforcement learning in finite MDPs: PAC analysis. *Journal of Machine Learning Research*, 10:2413–2444, 2009.

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

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

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

Sutton, R. S., Maei, H. R., Precup, D., Bhatnagar, S., Silver, D., Szepesvri, C., and Wiewiora, E. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *ICML*, pp. 993–1000, 2009.

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

Todorov, E., Erez, T., and Tassa, Y. 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, J. N. and Van Roy, B. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 42:674–690, 1997.

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

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

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

Appendix

A. Properties of Smoothed Bellman Operator

After applying the smoothing technique (Nesterov, 2005), we obtain a new Bellman operator, $\widetilde{\mathcal{T}}$, which is contractive. By such property, we can guarantee the uniqueness of the solution; a similar result is also presented in 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^*.*

Proof For any $V_1, V_2 : \mathcal{S} \to \mathbb{R}$, we have

$$\begin{aligned}
& \left\|\widetilde{\mathcal{T}}V_1 - \widetilde{\mathcal{T}}V_2\right\|_\infty \\
= & \left\|\max_\pi \left\{\left\langle \pi, R(s,a) + \gamma \mathbb{E}_{s'|s,a}\left[V_1(s')\right]\right\rangle + \lambda H(\pi)\right\} - \max_\pi \left\{\left\langle \pi, R(s,a) + \gamma \mathbb{E}_{s'|s,a}\left[V_2(s')\right]\right\rangle + \lambda H(\pi)\right\}\right\|_\infty \\
\leqslant & \left\|\max_\pi \left\{\left\langle \pi, R(s,a) + \gamma \mathbb{E}_{s'|s,a}\left[V_1(s')\right]\right\rangle + \lambda H(\pi) - \left\langle \pi, R(s,a) + \gamma \mathbb{E}_{s'|s,a}\left[V_2(s')\right]\right\rangle - \lambda H(\pi)\right\}\right\| \\
\leqslant & \left\|\max_\pi \left\langle \pi, \gamma \mathbb{E}_{s'|s,a}\left[V_1(s') - V_2(s')\right]\right\rangle\right\|_\infty \\
\leqslant & \gamma \left\|V_1 - V_2\right\|_\infty .
\end{aligned}$$

$\mathcal{T}_\lambda$ is therefore a γ-contraction and, by the Banach fixed point theorem, admits a unique fixed point. ■

Moreover, we may characterize the bias introduced by the entropic smoothing, similar to the simulation lemma (see, e.g., Kearns & Singh (2002) and Strehl et al. (2009)):

Proposition 2 (Smoothing bias) *Let V^* and V_λ^* be the fixed points of (2) and (4), respectively. It holds that*

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

As $\lambda \to 0$, V_λ^* converges to V^* pointwisely.

Proof Using the triangle inequality and the contraction property of $\mathcal{T}_\lambda$, we have

$$\begin{aligned}
\left\|V^* - V_\lambda^*\right\|_\infty &= \left\|\mathcal{T}V^* - \mathcal{T}_\lambda V_\lambda^*\right\|_\infty \\
&= \left\|V^* - \mathcal{T}_\lambda V^* + \mathcal{T}_\lambda V^* - \mathcal{T}_\lambda V_\lambda^*\right\|_\infty \\
&\leqslant \left\|V^* - \mathcal{T}_\lambda V^*\right\|_\infty + \left\|\mathcal{T}_\lambda V^* - \mathcal{T}_\lambda V_\lambda^*\right\|_\infty \\
&\leqslant \lambda H^* + \gamma \left\|V^* - V_\lambda^*\right\|_\infty ,
\end{aligned}$$

which immediately implies the desired bound. ■

The smoothed Bellman equation involves a log-sum-exp operator to approximate the max-operator, which increases the nonlinearity of the equation. We further characterize the solution of the smoothed Bellman equation, by the temporal consistency conditions.

Theorem 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, $(V, \pi) = (V_\lambda^*, \pi_\lambda^*)$ if and only if (V, π) satisfies the following equality for all $(s, a) \in \mathcal{S} \times \mathcal{A}$:*

$$V(s) = R(s,a) + \gamma \mathbb{E}_{s'|s,a}\left[V(s')\right] - \lambda \log \pi(a|s) . \tag{7}$$

Proof The proof has two parts.
(Necessity) We need to show $(V_\lambda^*, \pi_\lambda^*)$ is a solution to (6). Simple calculations give the closed form of π_λ^*:

$$\pi_\lambda^*(a|s) = Z(s)^{-1} \exp\left(\frac{R(s,a) + \gamma \mathbb{E}_{s'|s,a}\left[V_\lambda^*(s')\right]}{\lambda}\right) ,$$

where $Z(s) := \sum_{a\in\mathcal{A}} \exp\left(\frac{R(s,a)+\gamma\mathbb{E}_{s'|s,a}\left[V^*_\lambda(s')\right]}{\lambda}\right)$ is a state-dependent normalization constant. Therefore, for any $a \in \mathcal{A}$,

$$\begin{aligned} & R(s,a) + \gamma\mathbb{E}_{s'|s,a}\left[V^*_\lambda(s')\right] - \lambda \log \pi^*_\lambda(a|s) \\ = \;& R(s,a) + \gamma\mathbb{E}_{s'|s,a}\left[V^*_\lambda(s')\right] - \lambda\left(\frac{R(s,a)+\gamma\mathbb{E}_{s'|s,a}\left[V^*_\lambda(s')\right]}{\lambda} - \log Z(s)\right) \\ = \;& \lambda \log Z(s) \;=\; V^*_\lambda(s)\,, \end{aligned}$$

where the last step is from (5). Therefore, $(V^*_\lambda, \pi^*_\lambda)$ satisfies (6).

(Sufficiency) Assume $\bar{V}$ and $\bar{\pi}$ satisfies (6), then we have for all $(s,a) \in \mathcal{S}\times\mathcal{A}$ that

$$\begin{aligned} \bar{V}(s) &= R(s,a) + \gamma\mathbb{E}_{s'|s,a}\left[\bar{V}(s')\right] - \lambda\log\bar{\pi}(a|s) \\ \pi(a|s) &= \exp\left(\frac{R(s,a)+\gamma\mathbb{E}_{s'|s,a}\left[\bar{V}(s')\right]-\bar{V}(s)}{\lambda}\right). \end{aligned}$$

Recall $\pi(\cdot|s) \in \mathcal{P}$, we have

$$\begin{aligned} & \sum_{a\in\mathcal{A}} \exp\left(\frac{R(s,a)+\gamma\mathbb{E}_{s'|s,a}\left[\bar{V}(s')\right]-\bar{V}(s)}{\lambda}\right) = 1 \\ \Rightarrow \quad & \sum_{a\in\mathcal{A}} \exp\left(\frac{R(s,a)+\gamma\mathbb{E}_{s'|s,a}\left[\bar{V}(s')\right]}{\lambda}\right) = \exp\left(\frac{\bar{V}(s)}{\lambda}\right) \\ \Rightarrow \quad & \bar{V}(s) = \lambda\log\left(\sum_{a\in\mathcal{A}} \exp\left(\frac{R(s,a)+\gamma\mathbb{E}_{s'|s,a}\left[\bar{V}(s')\right]}{\lambda}\right)\right) = \mathcal{T}_\lambda\bar{V}(s)\,. \end{aligned}$$

The last equation holds for all $s \in \mathcal{S}$, so $\bar{V}$ is a fixed point of $\mathcal{T}$. It then follows from Proposition 1 that $\bar{V} = V^*_\lambda$. Finally, $\bar{\pi} = \pi^*_\lambda$ due to strong concavity of the entropy function ■

The same conditions have been re-discovered several times, *e.g.*, (Rawlik et al., 2012; Nachum et al., 2017), from a completely different point of views.

B. Variance Cancellation via the Saddle Point Formulation

The second term in the saddle point formulation (9) will cancel the variance $\mathbb{V}_{s,a,s'}\left[\gamma V(s')\right]$. Formally,

Proposition 8 *Given any fixed (V,π), we have*

$$\max_{\rho\in\mathcal{F}(\mathcal{S}\times\mathcal{A})} -\mathbb{E}_{s,a,s'}\left[\left(R(s,a)+\gamma V(s') - \lambda\log\pi(a|s) - \rho(s,a)\right)^2\right] = -\gamma^2\mathbb{E}_{s,a}\left[\mathbb{V}_{s'|s,a}\left[V(s')\right]\right]. \tag{11}$$

Proof Recall from (9) that $\delta(s,a,s') = R(s,a)+\gamma V(s') - \lambda\log\pi(a|s)$. Then,

$$\begin{aligned} & \max_\rho \; -\mathbb{E}_{s,a,s'}\left[\left(R(s,a)+\gamma V(s') - \lambda\log\pi(a|s) - \rho(s,a)\right)^2\right] \\ = \;& -\min_\rho \; \mathbb{E}_{s,a}\left[\mathbb{E}_{s'|s,a}\left[\left(\delta(s,a,s') - \rho(s,a)\right)^2\right]\right]. \end{aligned}$$

Clearly, the minimizing function ρ^* may be determined for each (s,a) entry separately. Fix any $(s,a) \in \mathcal{S}\times\mathcal{A}$, and define a function on $\mathbb{R}$ as $q(x) := \mathbb{E}_{s'|s,a}\left[\left(\delta(s,a,s')-x\right)^2\right]$. Obviously, this convex function is minimized at the stationary point $x^* = \mathbb{E}_{s'|s,a}\left[\delta(s,a,s')\right]$. We therefore have $\rho^*(s,a) = \mathbb{E}_{s'|s,a}\left[\delta(s,a,s')\right]$ for all (s,a), so

$$\begin{aligned} & \min_\rho \; \mathbb{E}_{s,a}\left[\mathbb{E}_{s'|s,a}\left[\left(\delta(s,a,s') - \rho(s,a)\right)^2\right]\right] \\ = \;& \mathbb{E}_{s,a}\left[\mathbb{E}_{s'|s,a}\left[\left(\delta(s,a,s') - \mathbb{E}_{s'|s,a}\left[\delta(s,a,s')\right]\right)^2\right]\right] \\ = \;& \mathbb{E}_{s,a}\left[\mathbb{V}_{s'|s,a}\left[\delta(s,a,s')\right]\right] \\ = \;& \mathbb{E}_{s,a}\left[\mathbb{V}_{s'|s,a}\left[\gamma V(s')\right]\right] \;=\; \gamma^2\mathbb{E}_{s,a}\left[\mathbb{V}_{s'|s,a}\left[V(s')\right]\right], \end{aligned}$$

where the second last step is due to the fact that, conditioned on s and a, the only random variable in $\delta(s,a,s')$ is $V(s')$. ■

C. Details of SBEED

In this section, we provide further details of the SBEED algorithms, including its gradient derivation and multi-step/eligibility-trace extension.

C.1. Unbiasedness of Gradient Estimator

In this subsection, we compute the gradient with respect to the primal variables. Let (w_V, w_π) be the parameters of the primal (V, π), and w_ρ the parameters of the dual ρ. Abusing notation a little bit, we now write the objective function $L_\eta(V, \pi; \rho)$ as $L_\eta(w_V, w_\pi; w_\rho)$. Recall the quantity $\delta(s, a, s')$ from (9).

Theorem 4 (Gradient derivation) *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,*

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

Proof First, note that w_ρ^* is an implicit function of (w_V, w_π). Therefore, we must use the chain rule to compute the gradient:

$$\begin{aligned}\nabla_{w_V} \bar{\ell}_\eta \;=\;& 2\mathbb{E}_{s,a,s'}\left[(\delta_{s,a,s'} - V(s;w_V))\left(\gamma \nabla_{w_V} V(s';w_V) - \nabla_{w_V} V(s;w_V)\right)\right]\\ & -2\eta\gamma \mathbb{E}_{s,a,s'}\left[\left(\delta_{s,a,s'} - \rho(s,a;w_\rho^*)\right)\nabla_{w_V} V(s';w_V)\right]\\ & +2\eta\gamma \mathbb{E}_{s,a,s'}\left[\left(\delta_{s,a,s'} - \rho(s,a;w_\rho^*)\right)\nabla_{w_V} \rho(s,a;w_\rho^*)\right] .\end{aligned}$$

We next show that the last term is zero:

$$\begin{aligned}& \mathbb{E}_{s,a,s'}\left[\left(\delta_{s,a,s'} - \rho(s,a;w_\rho^*)\right)\nabla_{w_V} \rho(s,a;w_\rho^*)\right]\\ =\;& \mathbb{E}_{s,a,s'}\left[\left(\delta_{s,a,s'} - \rho(s,a;w_\rho^*)\right)\cdot \nabla_{w_V} w_\rho^* \cdot \nabla_{w_\rho} \rho(s,a;w_\rho^*)\right]\\ =\;& \nabla_{w_V} w_\rho^* \cdot \mathbb{E}_{s,a,s'}\left[\left(\delta_{s,a,s'} - \rho(s,a;w_\rho^*)\right)\cdot \nabla_{w_\rho} \rho(s,a;w_\rho^*)\right]\\ =\;& \nabla_{w_V} w_\rho^* \cdot \mathbf{0} \;=\; \mathbf{0} ,\end{aligned}$$

where the first step is the chain rule; the second is due to the fact that $\nabla_{w_V} w_\rho^*$ is not a function of (s, a, s'), so can be moved outside of the expectation; the third step is due to the optimality of w_ρ^*. The gradient w.r.t. w_V is thus derived. The case for w_π is similar. ■

C.2. Multi-step Extension

One way to interpret the smoothed Bellman equation (4) is to treat each $\pi(\cdot|s)$ as a (mixture) action; in other words, the action space is now the simplex $\mathcal{P}_\mathcal{A}$. With this interpretation, the introduced entropy regularization may be viewed as a shaping reward: given a mixture action $\pi(\cdot|s)$, its immediate reward is given by

$$\tilde{R}(s, \pi(\cdot|s)) := \mathbb{E}_{a\sim\pi(\cdot|s)}\left[R(s,a)\right] + \lambda H(\pi, s) .$$

The transition probabilities can also be adapted accordingly as follows

$$\tilde{P}(s'|s, \pi(\cdot|s)) := \mathbb{E}_{a\in\pi(\cdot|s)}\left[P(s'|s,a)\right] .$$

It can be verified that the above constructions induce a well-defined MDP $\tilde{M} = \langle \mathcal{S}, \mathcal{P}_\mathcal{A}, \tilde{P}, \tilde{R}, \gamma\rangle$, whose standard Bellman equation is exactly (4).

With this interpretation, the proposed framework and algorithm can be easily applied to multi-step and eligibility-traces extensions. Specifically, one can show that $(V_\lambda^*, \pi_\lambda^*)$ is the unique solution that satisfies the multi-step expansion of (6): for any $k \geqslant 1$ and any $(s_0, a_0, a_1, \ldots, a_{k-1}) \in \mathcal{S}\times\mathcal{A}^k$,

$$V(s_0) = \sum_{t=0}^{k-1} \gamma^t \mathbb{E}_{s_t|s_0,a_{0:t-1}}\left[R(s_t,a_t) - \lambda \log \pi(a_t|s_t)\right] + \gamma^k \mathbb{E}_{s_k|s_0,a_{0:k-1}}\left[V(s_k)\right] . \tag{12}$$

Clearly, when $k = 1$ (the single-step bootstrapping case), the above equation reduces to (6).

The k-step extension of objective function (7) now becomes

$$\min_{V,\pi} \mathbb{E}_{s_0,a_{0:k-1}} \left[\left(\sum_{t=0}^{k-1} \gamma^t \mathbb{E}_{s_t|s_0,a_{0:t-1}} \left[R(s_t,a_t) - \lambda \log \pi(a_t|s_t) \right] + \gamma^k \mathbb{E}_{s_k|s_0,a_{0:k-1}} \left[V(s_k) \right] - V(s_0) \right)^2 \right].$$

Applying the Legendre-Fenchel transformation and the interchangeability principle, we arrive at the following multi-step primal-dual optimization problem:

$$\begin{aligned} \min_{V,\pi} \max_{\nu} \quad & \mathbb{E}_{s_0,a_{0:t-1}} \Bigg[\nu(s_0,a_{0:t-1}) \Bigg(\sum_{t=0}^{k-1} \gamma^t \mathbb{E}_{s_t|s_0,a_{0:k-1}} \left[R(s_t,a_t) - \lambda \log \pi(a_t|s_t) \right] \\ & + \gamma^k \mathbb{E}_{s_k|s_0,a_{0:k-1}} \left[V(s_k) \right] - V(s_0) \Bigg) \Bigg] - \frac{1}{2} \mathbb{E}_{s_0,a_{0:k-1}} \left[\nu(s_0,a_{0:k-1})^2 \right] \\ = \min_{V,\pi} \max_{\nu} \quad & \mathbb{E}_{s_{0:k},a_{0:k-1}} \Bigg[\nu(s_0,a_{0:k-1}) \Bigg(\sum_{t=0}^{k-1} \gamma^t \left(R(s_t,a_t) - \lambda \log \pi(a_t|s_t) \right) \\ & + \gamma^k V(s_k) - V(s_0) \Bigg) \Bigg] - \frac{1}{2} \mathbb{E}_{s_0,a_{0:k-1}} \left[\nu(s_0,a_{0:k-1})^2 \right]. \end{aligned}$$

Similar to the single-step case, defining

$$\delta(s_{0:k},a_{0:k-1}) := \sum_{t=0}^{k-1} \gamma^t \left(R(s_t,a_t) - \lambda \log \pi(a_t|s_t) \right) + \gamma^k V(s_k).$$

and using the substitution $\rho(s_0,a_{0:k-1}) = \nu(s_0,a_{0:k-1}) + V(s_0)$, we reach the following saddle-point formulation:

$$\min_{V,\pi} \max_{\rho} L(V,\pi;\rho) := \mathbb{E}_{s_{0:k},a_{0:k-1}} \left[\left(\delta(s_{0:k},a_{0:k-1}) - V(s_0) \right)^2 - \eta \left(\delta(s_{0:k},a_{0:k-1}) - \rho(s_0,a_{0:k-1}) \right)^2 \right] \tag{13}$$

where the dual function now is $\rho(s_0,a_{0:k-1})$, a function on $\mathcal{S} \times \mathcal{A}^k$, and $\eta \geqslant 0$ is again a parameter used to balance between bias and variance. It is straightforward to generalize Theorem 4 to the multi-step setting, and to adapt SBEED accordingly,

C.3. Eligibility-trace Extension

Eligibility traces can be viewed as an approach to aggregating multi-step bootstraps for $k \in \{1,2,\cdots\}$; see Sutton & Barto (1998) for more discussions. The same can be applied to the multi-step consistency condition (12), using an exponential weighting parameterized by $\zeta \in [0,1)$. Specifically, for all $(s_0,a_{0:k-1}) \in \mathcal{S} \times \mathcal{A}^k$, we have

$$V(s_0) = (1-\zeta) \sum_{k=1}^{\infty} \zeta^{k-1} \left(\sum_{t=0}^{k-1} \gamma^t \mathbb{E}_{s_t|s_0,a_{0:k-1}} \left[R(s_t,a_t) - \lambda \log \pi(a_t|s_t) \right] + \gamma^k \mathbb{E}_{s_k|s_0,a_{0:k-1}} \left[V(s_k) \right] \right). \tag{14}$$

Then, following similar steps as in the previous subsection, we reach the following saddle-point optimization:

$$\begin{aligned} \min_{V,\pi} \max_{\rho} \quad & \mathbb{E}_{s_{0:\infty},a_{0:\infty}} \left[\left((1-\zeta) \sum_{k=1}^{\infty} \zeta^{k-1} \delta(s_{0:k},a_{0:k-1}) - V(s_0) \right)^2 \right] \\ & - \eta \, \mathbb{E}_{s_{0:\infty},a_{0:\infty}} \left[\left((1-\zeta) \sum_{k=1}^{\infty} \zeta^{k-1} \delta(s_{0:k},a_{0:k-1}) - \rho(s_0,a_{0:\infty}) \right)^2 \right]. \end{aligned} \tag{15}$$

In practice, $\rho(s_0,a_{0:\infty})$ can be parametrized by neural networks with finite length of actions as input as an approximation.

D. Proof Details of the Theoretical Analysis

In this section, we provide the details of the analysis in Theorems 6 and 7. We start with the boundedness of V^* and V_λ^* under Assumption 1. Given any measure on the state space $\mathcal{S}$,

$$\|V^*\|_\mu \leqslant \|V^*\|_\infty \leqslant (1+\gamma+\gamma^2+\cdots) C_R = C_V := \frac{C_R}{1-\gamma}.$$

A similar argument may be used on V_λ^* to get

$$\|V_\lambda^*\|_\mu \leqslant \frac{C_R + H^*}{1-\gamma}.$$

It should be emphasized that although Assumption 1 ensures boundedness of V^* and $\log \pi^*(a|s)$, it does not imply the continuity and smoothness. In fact, as we will see later, λ controls the trade-off between approximation error (due to parameterization) and bias (due to smoothing) in the solution of the smoothed Bellman equation.

D.1. Error Decomposition

Recall that

- (V^*, π^*) corresponds to the optimal value function and optimal policy to the original Bellman equation, namely, they are solutions to the optimization problem (3);
- $(V_\lambda^*, \pi_\lambda^*)$ corresponds to the optimal value function and optimal policy to the smoothed Bellman equation, namely, they are solutions to the optimization problem (7) with objective $\ell(V, \pi)$;
- (V_w^*, π_w^*) correponds to the optimal solution to the optimization problem (7) under nonlinear function approximation, with objective $\ell_w(V_w, \pi_w)$;
- $(\widehat{V}_w^*, \widehat{\pi}_w^*)$ stands for the optimal solution to the finite sample approximation of (7) under nonlinear function approximation, with objective $\widehat{\ell}_T(V_w, \pi_w)$.

Hence, we can decompose the error between $(\widehat{V}_w^*, \widehat{\pi}_w^*)$ and $(V_\lambda^*, \pi_\lambda^*)$ under the $\|\cdot\|_{\mu\pi_b}$ norm.

$$\left\|\widehat{V}_w^* - V^*\right\|_{\mu\pi_b}^2 \leqslant 2\left\|\widehat{V}_w^* - V_\lambda^*\right\|_{\mu\pi_b}^2 + 2\left\|V_\lambda^* - V^*\right\|_{\mu\pi_b}^2. \tag{16}$$

We first look at the second term from smoothing error, which can be similarly bounded, as shown in Proposition 2.

Lemma 9 (Smoothing bias) $\|V_\lambda^* - V^*\|_{\mu\pi_b}^2 \leqslant (2\gamma^2 + 2)\left(\frac{\gamma\lambda}{1-\gamma}\max_{\pi\in\mathcal{P}} H(\pi)\right)^2$.

Proof For $\|V_\lambda^* - V^*\|_{\mu\pi_b}^2$, we have

$$\begin{aligned}
\|V_\lambda^* - V^*\|_{\mu\pi_b}^2 &= \int \left(\gamma\mathbb{E}_{s'|s,a}\left[V^*(s') - V_\lambda^*(s')\right] - \left(V^*(s) - V_\lambda^*(s)\right)\right)^2 \mu(s)\pi_b(a|s)dsda \\
&\leqslant 2\gamma^2\left\|\mathbb{E}_{s'|s,a}\left[V^*(s') - V_\lambda^*(s')\right]\right\|_\infty^2 + 2\left\|V^*(s) - V_\lambda^*(s)\right\|_\infty \\
&\leqslant (2\gamma^2+2)\left(\frac{\gamma\lambda}{1-\gamma}\max_{\pi\in\mathcal{P}} H(\pi)\right)^2,
\end{aligned}$$

where the final inequality is because Lemma 2. ■

We now look at the first term and show that

Lemma 10

$$\begin{aligned}
\left\|\widehat{V}_w^* - V_\lambda^*\right\|_{\mu\pi_b}^2 \leqslant\ & 2\left(\ell(\widehat{V}_w^*, \widehat{\pi}_w^*) - \ell(V_\lambda^*, \pi_\lambda^*)\right) + 4\lambda^2\left\|\log\widehat{\pi}_w^*(a|s) - \log\hat{\pi}_w^*(a|s)\right\|_2^2 \\
& + 4\lambda^2\left\|\log\pi_w^*(a|s) - \log\pi_\lambda^*(a|s)\right\|_2^2.
\end{aligned}$$

Proof Specifically, due to the strongly convexity of square function, we have

$$\begin{aligned}
\ell(\widehat{V}_w^*, \widehat{\pi}_w^*) - \ell(V_\lambda^*, \pi_\lambda^*) &= 2\mathbb{E}\left[\bar{\Delta}_{V_\lambda^*,\pi_\lambda^*}(s,a)\left(\bar{\Delta}_{\widehat{V}_w^*,\widehat{\pi}_w^*}(s,a) - \bar{\Delta}_{V_\lambda^*,\pi_\lambda^*}(s,a)\right)\right] \\
&+ \mathbb{E}_{\mu\pi_b}\left[\left(\bar{\Delta}_{\widehat{V}_w^*,\widehat{\pi}_w^*}(s,a) - \bar{\Delta}_{V_\lambda^*,\pi_\lambda^*}(s,a)\right)^2\right] \\
&\geqslant \int\left(\bar{\Delta}_{\widehat{V}_w^*,\widehat{\pi}_w^*}(s,a) - \bar{\Delta}_{V_\lambda^*,\pi_\lambda^*}(s,a)\right)^2\mu(s)\pi_b(a|s)dsda \\
&:= \left\|\bar{\Delta}_{\widehat{V}_w^*,\widehat{\pi}_w^*}(s,a) - \bar{\Delta}_{V_\lambda^*,\pi_\lambda^*}(s,a)\right\|_2^2,
\end{aligned}$$

where $\Delta(s,a,s') = R(s,a) + \gamma V(s') - \lambda \log \pi(a|s) - V(s)$ and the second inequality is because the optimality of V_λ^* and π_λ^*. Therefore, we have

$$\begin{aligned}
&\sqrt{\ell(\widehat{V}_w^*, \widehat{\pi}_w^*) - \ell(V_\lambda^*, \pi_\lambda^*)} \geqslant \left\| \bar{\Delta}_{\widehat{V}_w^*, \widehat{\pi}_w^*}(s,a) - \bar{\Delta}_{V_\lambda^*, \pi_\lambda^*}(s,a) \right\|_2 \\
\geqslant & \left| \left\| \gamma \mathbb{E}_{s'|s,a} \left[\widehat{V}_w^*(s') - V_\lambda^*(s') \right] - \left(\widehat{V}_w^*(s) - V_\lambda^*(s) \right) \right\|_2 - \lambda \left\| \log \widehat{\pi}_w^*(a|s) - \log \pi_\lambda^*(a|s) \right\|_2 \right| \\
= & \left| \left\| \widehat{V}_w^* - V_\lambda^* \right\|_{\mu\pi_b} - \lambda \left\| \log \widehat{\pi}_w^*(a|s) - \log \pi_\lambda^*(a|s) \right\|_2 \right|
\end{aligned}$$

which implies

$$\begin{aligned}
\left\| \widehat{V}_w^* - V_\lambda^* \right\|_{\mu\pi_b}^2 \leqslant & \; 2 \left(\ell(\widehat{V}_w^*, \widehat{\pi}_w^*) - \ell(V_\lambda^*, \pi_\lambda^*) \right) + 2\lambda^2 \left\| \log \widehat{\pi}_w^*(a|s) - \log \pi_\lambda^*(a|s) \right\|_2^2 \\
\leqslant & \; 2 \left(\ell(\widehat{V}_w^*, \widehat{\pi}_w^*) - \ell(V_\lambda^*, \pi_\lambda^*) \right) + 4\lambda^2 \left\| \log \widehat{\pi}_w^*(a|s) - \log \pi_w^*(a|s) \right\|_2^2 \\
& + 4\lambda^2 \left\| \log \pi_w^*(a|s) - \log \pi_\lambda^*(a|s) \right\|_2^2 .
\end{aligned}$$

■

In regular MDP with Assumption 1, with appropriate C, such constraint does not introduce any loss. We denote the family of value functions and policies by parametrization as $\mathcal{V}_w$, $\mathcal{P}_w$, respectively. Then, for V and $\log \pi$ uniformly bounded by $C_\infty = \max\left\{ \frac{C_R}{1-\gamma}, C_\pi \right\}$ and the square loss is uniformly K-Lipschitz continuous, by proposition in Dai et al. (2017), we have

Corollary 11 *$\ell(V,\pi) - \ell_w(V,\pi) \leqslant (K + C_\infty)\epsilon_{app}^\nu$ where $\epsilon_{app} = \sup_{\nu \in \mathcal{C}} \inf_{h \in \mathcal{H}} \|\nu - h\|_\infty$ with $\mathcal{C}$ denoting the Lipschitz continuous function space and $\mathcal{H}$ denoting the hypothesis space.*

Proof Denote the $\phi(V,\pi,\nu) := \mathbb{E}_{s,a,s'}\left[\nu(s,a)\left(R(s,a) + \gamma V(s') - \lambda \log \pi(a|s) - V(s)\right)\right] - \frac{1}{2}\mathbb{E}_{s,a,s'}\left[\nu^2(s,a)\right]$, we have $\phi(V,\pi,\nu)$ is $(K + C_\infty)$-Lipschitz continuous w.r.t. $\|\cdot\|_\infty$. Denote $\nu_{V,\pi}^* = \operatorname{argmax}_\nu \phi(V,\pi,\nu)$, $\nu_{V,\pi}^{\mathcal{H}} = \operatorname{argmax}_{\nu \in \mathcal{H}} \phi(V,\pi,\nu)$, and $\hat{\nu}_{V,\pi} = \min_{\nu \in \mathcal{H}} \left\| \nu - \nu_{V,\pi}^* \right\|_\infty$

$$\begin{aligned}
\ell(V,\pi) - \ell_w(V,\pi) &= \phi(V,\pi,\nu_{V,\pi}^*) - \phi(V,\pi,\nu_{V,\pi}^{\mathcal{H}}) \\
&\leqslant \phi(V,\pi,\nu_{V,\pi}^*) - \phi(V,\pi,\hat{\nu}_{V,\pi}) \leqslant (K + C_\infty)\epsilon_{\text{app}}^\nu.
\end{aligned}$$

■

For the third term in Lemma 10, we have

$$\begin{aligned}
\lambda \left\| \log \pi_w^*(a|s) - \log \pi_\lambda^*(a|s) \right\|_2^2 &\leqslant \ell\left(V, \pi_w^*\right) - \ell\left(V, \pi_\lambda^*\right) \\
&= \ell_w\left(V, \pi_w^*\right) - \ell_w\left(V, \pi_\lambda^*\right) + \left(\ell\left(V, \pi_w^*\right) - \ell_w\left(V, \pi_w^*\right)\right) - \left(\ell\left(V, \pi_\lambda^*\right) - \ell_w\left(V, \pi_\lambda^*\right)\right) \\
&\leqslant C_\nu \inf_{\pi_w} \left\| \lambda \log \pi_w - \lambda \log \pi_\lambda^* \right\|_\infty + 2(K + C_\infty)\epsilon_{\text{app}}^\nu \\
&\leqslant C_\nu \epsilon_{\text{app}}^\pi(\lambda) + 2(K + C_\infty)\epsilon_{\text{app}}^\nu
\end{aligned} \tag{17}$$

where $C_\nu = \max_{\nu \in \mathcal{H}_w} \|\nu\|_2$. The first inequality comes from the strongly convexity of $\ell\left(V,\pi\right)$ w.r.t. $\lambda \log \pi$, the second inequality comes from Section 5 in Bach (2014) and Corollary 11 with $\epsilon_{\text{app}}^\pi(\lambda) := \sup_{\pi \in \mathcal{P}_\lambda} \inf_{\pi_w \in \mathcal{P}_w} \left\| \lambda \log \pi_w - \lambda \log \pi \right\|_\infty$ with

$$\mathcal{P}_\lambda := \left\{ \pi \in \mathcal{P}, \pi(a|s) = \exp\left(\frac{Q(s,a) - \mathcal{L}(Q)}{\lambda} \right), \|Q\|_2 \leqslant C_V \right\}.$$

Based on the derivation of $\mathcal{P}_\lambda$, with continuous $\mathcal{A}$, it can be seen that as $\lambda \to 0$,

$$\mathcal{P}_0 = \left\{ \pi \in \mathcal{P}, \pi(a|s) = \delta_{a_{\max}(s)}(a) \right\},$$

which results $\epsilon_{\text{app}}^\pi(\lambda) \to \infty$, and as λ increasing as finite, the policy becomes smoother, resulting smaller approximate error in general. With discrete $\mathcal{A}$, although the $\epsilon_{\text{app}}^\pi(0)$ is bounded, the approximate error still decreases as λ increases. The similar correspondence also applies to $\epsilon_{\text{app}}^V(\lambda)$. The concrete correspondence between λ and $\epsilon_{\text{app}}(\lambda)$ depends on the specific form of the function approximators, which is an open problem and out of the scope of this paper.

For the second term in 10,

$$\lambda \left\|\log \widehat{\pi}_w^*(a|s) - \log \pi_w^*(a|s)\right\|_2 \leqslant \lambda \left\|\log \widehat{\pi}_w^*(a|s)\right\|_2 + \lambda \left\|\log \pi_w^*(a|s)\right\|_2 \leqslant 2\lambda C_\pi. \tag{18}$$

For the first term, we have

$$\begin{aligned} & \ell(\widehat{V}_w^*, \widehat{\pi}_w^*) - \ell(V_\lambda^*, \pi_\lambda^*) \\ = \;& \ell(\widehat{V}_w^*, \widehat{\pi}_w^*) - \ell_w(\widehat{V}_w^*, \widehat{\pi}_w^*) + \ell_w(\widehat{V}_w^*, \widehat{\pi}_w^*) - \ell_w(V_\lambda^*, \pi_\lambda^*) + \ell_w(V_\lambda^*, \pi_\lambda^*) - \ell(V_\lambda^*, \pi_\lambda^*) \\ \leqslant \;& 2(K + C_\infty)\epsilon_{\text{app}}^\nu + \ell_w(\widehat{V}_w^*, \widehat{\pi}_w^*) - \ell_w(V_\lambda^*, \pi_\lambda^*) \\ = \;& 2(K + C_\infty)\epsilon_{\text{app}}^\nu + \ell_w(\widehat{V}_w^*, \widehat{\pi}_w^*) - \ell_w(V_w^*, \pi_w^*) + \ell_w(V_w^*, \pi_w^*) - \ell_w(V_\lambda^*, \pi_\lambda^*) \\ \leqslant \;& 2(K + C_\infty)\epsilon_{\text{app}}^\nu + C_\nu \left((1+\gamma)\epsilon_{\text{app}}^V(\lambda) + \epsilon_{\text{app}}^\pi(\lambda)\right) + \ell_w(\widehat{V}_w^*, \widehat{\pi}_w^*) - \ell_w(V_w^*, \pi_w^*). \end{aligned} \tag{19}$$

The last inequality is because

$$\begin{aligned} \ell_w(V_w^*, \pi_w^*) - \ell_w(V_\lambda^*, \pi_\lambda^*) &= \inf_{V_w, \pi_w} \ell_w(V_w, \pi_w) - \ell_w(V_\lambda^*, \pi_\lambda^*) \\ &\leqslant C_\nu \inf_{V_w, \pi_w} \left((1+\gamma)\left\|V_w - V_\lambda^*\right\|_\infty + \lambda \left\|\log \pi_w - \log \pi_\lambda^*\right\|_\infty\right) \\ &\leqslant C_\nu \left((1+\gamma)\epsilon_{\text{app}}^V(\lambda) + \epsilon_{\text{app}}^\pi(\lambda)\right), \end{aligned}$$

where the second inequality comes from Section 5 in Bach (2014).

Combine (17), (18) and (19) into Lemma 10 and Lemma 9 together with (16), we achieve

Lemma 12 (Error decomposition)

$$\begin{aligned} \left\|\widehat{V}_w^* - V^*\right\|_{\mu\pi_b}^2 \leqslant\; & \underbrace{2\left(4(K + C_\infty)\epsilon_{app}^\nu + C_\nu(1+\gamma)\epsilon_{app}^V(\lambda) + 3C_\nu\epsilon_{app}^\pi(\lambda)\right)}_{\textit{approximation error due to parametrization}} \\ & \underbrace{+ 16\lambda^2 C_\pi^2 + \left(2\gamma^2 + 2\right)\left(\frac{\gamma\lambda}{1-\gamma}\max_{\pi\in\mathcal{P}} H(\pi)\right)^2}_{\textit{bias due to smoothing}} + \underbrace{2\left(\ell_w(\widehat{V}_w^*, \widehat{\pi}_w^*) - \ell_w(V_w^*, \pi_w^*)\right)}_{\textit{statistical error}}. \end{aligned}$$

We can see that the bound includes the errors from three aspects: **i)**, the approximation error induced by parametrization of V, π, and ν; **ii)**, the bias induced by smoothing technique; **iii)**, the statistical error. As we can see from Lemma 12, λ plays an important role in balance the approximation error and smoothing bias.

D.2. Statistical Error

In this section, we analyze the generalization error. For simplicity, we denote the T finite-sample approximation of

$$L(V, \pi, \nu) = \mathbb{E}[\phi_{V,\pi,\nu}(s, a, R, s')] := \mathbb{E}\left[2\nu(s,a)\left(R(s,a) + \gamma V(s') - V(s) - \lambda \log \pi(a|s)\right) - \nu^2(s,a)\right],$$

as

$$\widehat{L}_T(V, \pi, \nu) = \frac{1}{T}\sum_{i=1}^T \phi_{V,\pi,\nu}(s, a, R, s') := \frac{1}{T}\sum_{i=1}^T \left(2\nu(s_i,a_i)\left(R(s_i,a_i) + \gamma V(s_i') - V(s_i) - \lambda \log \pi(a_i|s_i)\right) - \nu^2(s_i,a_i)\right),$$

where the samples $\{(s_i, a_i, s_i', R_i)\}_{i=0}^T$ are sampled *i.i.d.* or from β-mixing stochastic process.

By definition, we have,

$$\begin{aligned} & \ell_w(\widehat{V}_w^*, \widehat{\pi}_w^*) - \ell_w(V_w^*, \pi_w^*) \\ = \;& \max_{\nu\in\mathcal{H}_w} L_w\left(\widehat{V}_w^*, \widehat{\pi}_w^*, \nu\right) - \max_{\nu\in\mathcal{H}_w} L_w\left(V_w^*, \hat{\pi}_w^*, \nu\right) \\ = \;& L_w\left(\widehat{V}_w^*, \widehat{\pi}_w^*, \nu_w\right) - L_w\left(V_w^*, \hat{\pi}_w^*, \nu_w\right) + \underbrace{L_w\left(V_w^*, \hat{\pi}_w^*, \nu_w\right) - \max_{\nu\in\mathcal{H}_w} L_w\left(V_w^*, \hat{\pi}_w^*, \nu\right)}_{\leqslant 0} \\ \leqslant \;& L_w\left(\widehat{V}_w^*, \widehat{\pi}_w^*, \nu_w\right) - L_w\left(V_w^*, \hat{\pi}_w^*, \nu_w\right) \\ \leqslant \;& 2\sup_{V,\pi,\nu\in\mathcal{F}_w\times\mathcal{P}_w\times\mathcal{H}_w} \left|\widehat{L}_T\left(V, \pi, \nu\right) - L_w\left(V, \pi, \nu\right)\right| \end{aligned}$$

where $\nu_w = \max_{\nu \in \mathcal{H}_w} L_w\left(\widehat{V}_w^*, \widehat{\pi}_w^*, \nu\right)$.

The latter can be bounded by covering number or Rademacher complexity on hypothesis space $\mathcal{F}_w \times \mathcal{P}_w \times \mathcal{H}_w$ with rate $\mathcal{O}\left(\sqrt{\frac{\log T}{T}}\right)$ with high probability if the samples are *i.i.d.* or from β-mixing stochastic processes (Antos et al., 2008).

We will use a generalized version of Pollard's tail inequality to β-mixing sequences, *i.e.*,

Lemma 13 *[Lemma 5, Antos et al. (2008)] Suppose that $z_1, \ldots, Z_N \in \mathcal{Z}$ is a stationary β-mixing process with mixing coefficient $\{\beta_m\}$ and that $\mathcal{G}$ is a permissible class of $\mathcal{Z} \to [-C, C]$ functions, then,*

$$\mathbb{P}\left(\sup_{g \in \mathcal{G}} \left|\frac{1}{N}\sum_{i=1}^{N} g(Z_i) - \mathbb{E}\left[g(Z_1)\right]\right| > \epsilon\right) \leqslant \quad 16\mathbb{E}\left[\mathcal{N}_1\left(\frac{\epsilon}{8}, \mathcal{G}, (Z_i'; i \in H)\right)\right] \exp\left(\frac{-m_N \epsilon^2}{128C^2}\right) + 2m_N \beta_{k_N+1},$$

where the "ghost" samples $Z_i' \in \mathcal{Z}$ and $H = \cup_{j=1}^{m_N} H_i$ which are defined as the blocks in the sampling path.

The covering number is highly related to pseudo-dimension, *i.e.*,

Lemma 14 *[Corollary 3, Haussler (1995)] For any set $\mathcal{X}$, any points $x^{1:N} \in \mathcal{X}^N$, any class $\mathcal{F}$ of functions on $\mathcal{X}$ taking values in $[0, C]$ with pseudo-dimension $D_{\mathcal{F}} < \infty$, and any $\epsilon > 0$,*

$$\mathcal{N}\left(\epsilon, \mathcal{F}, x^{1:N}\right) \leqslant e\left(D_{\mathcal{F}} + 1\right)\left(\frac{2eC}{\epsilon}\right)^{D_{\mathcal{F}}}$$

Once we have the covering number of $\Phi(V, \pi, \nu)$, plug it into lemma 13, we will achieve the statistical error,

Theorem 6 (Stochastic error) *Under Assumption 2, with at least probability $1 - \delta$,*

$$\ell_w(\widehat{V}_w^*, \widehat{\pi}_w^*) - \ell_w(V_w^*, \pi_w^*) \leqslant 2\sqrt{\frac{M\left(\max\left(M/b, 1\right)\right)^{1/\kappa}}{C_2 T}},$$

where $M = \frac{D}{2}\log t + \log\left(e/\delta\right) + \log^+\left(\max\left(C_1 C_2^{D/2}, \bar{\beta}\right)\right)$.

Proof We use lemma 13 with $\mathcal{Z} = \mathcal{S} \times \mathcal{A} \times \mathbb{R} \times \mathcal{S}$ and $\mathcal{G} = \phi_{\mathcal{F}_w \times \mathcal{P}_w \times \mathcal{H}_w}$. For $\forall \Phi(V, \pi, \nu) \in \mathcal{G}$, it is bounded by $C = \frac{2}{1-\gamma} C_R + \lambda C_\pi$. Thus,

$$\mathbb{P}\left(\sup_{V, \pi, \nu \in \mathcal{F}_w \times \mathcal{P}_w \times \mathcal{H}_w} \left|\frac{1}{T}\sum_{i=1}^{T} \phi_{V,\pi,\nu}\left((s, a, s', R)_i\right) - \mathbb{E}\left[\phi_{V,\pi,\nu}\right]\right| \geqslant \epsilon/2\right) \tag{20}$$

$$\leqslant 16\mathbb{E}\left[\mathcal{N}\left(\frac{\epsilon}{16}, \mathcal{G}, (Z_i'; i \in H)\right)\right] \exp\left(-\frac{m_t}{2}\left(\frac{\epsilon^2}{16C}\right)^2\right) + 2m_T \beta_{k_T}. \tag{21}$$

With some calculation, the distance in $\mathcal{G}$ can be bounded,

$$\begin{aligned} & \frac{1}{T}\sum_{i \in H} \left|\phi_{V_1,\pi_1,\nu_1}(Z_i') - \phi_{V_2,\pi_2,\nu_2}(Z_i')\right| \\ \leqslant \quad & \frac{4C}{T}\sum_{i \in H} \left|\nu_1(s_i, a_i) - \nu_2(s_i, a_i)\right| + \frac{2(1+\gamma)C}{T}\sum_{i \in H} \left|V_1(s_i) - V_2\left(s_i\right)\right| \\ & + \frac{2\lambda C}{T}\sum_{i \in H} \left|\log \pi_1(a_i|s_i) - \log \pi_2(a_i|s_i)\right|, \end{aligned}$$

which leads to

$$\mathcal{N}\left(12C\epsilon', \mathcal{G}, (Z_i'; i \in H)\right) \leqslant \mathcal{N}(\epsilon', \mathcal{F}_w, (Z_i'; i \in H))\mathcal{N}(\epsilon', \mathcal{P}_w, (Z_i'; i \in H))\mathcal{N}(\epsilon', \mathcal{H}_w, (Z_i'; i \in H))$$

with $\lambda \in (0, 2]$. To bound these factors, we apply lemma 14. We denote the psuedo-dimension of $\mathcal{F}_w$, $\mathcal{P}_w$, and $\mathcal{H}_w$ as D_V, D_π, and D_ν, respectively. Thus,

$$\mathcal{N}\left(12C\epsilon', \mathcal{G}, (Z_i'; i \in H)\right) \leqslant e^3\left(D_V + 1\right)\left(D_\pi + 1\right)\left(D_\nu + 1\right)\left(\frac{4eC}{\epsilon'}\right)^{D_V + D_\pi + D_\nu},$$

which implies

$$\mathcal{N}\left(\frac{\epsilon}{16}, \mathcal{G}, (Z_i'; i \in H)\right) \leqslant e^3 (D_V+1)(D_\pi+1)(D_\nu+1)\left(\frac{768eC^2}{\epsilon'}\right)^{D_V+D_\pi+D_\nu} = C_1\left(\frac{1}{\epsilon}\right)^D,$$

where $C_1 = e^3 (D_V+1)(D_\pi+1)(D_\nu+1)\left(768eC^2\right)^D$ and $D = D_V + D_\pi + D_\nu$, *i.e.*, the "effective" psuedo-dimension.

Plug this into Eq. (20), we obtain

$$\begin{aligned}
&\mathbb{P}\left(\sup_{V,\pi,\nu \in \mathcal{F}_w \times \mathcal{P}_w \times \mathcal{H}_w} \left| \frac{1}{T}\sum_{i=1}^T \phi_{V,\pi,\nu}((s,a,s',R)_i) - \mathbb{E}\left[\phi_{V,\pi,\nu}\right] \right| \geqslant \epsilon/2 \right) \\
\leqslant\ & C_1\left(\frac{1}{\epsilon}\right)^D \exp\left(-4C_2 m_t \epsilon^2\right) + 2m_T \beta_{k_T},
\end{aligned}$$

with $C_2 = \frac{1}{2}\left(\frac{1}{8C}\right)^2$. If $D \geqslant 2$, and $C_1, C_2, \bar{\beta}, b, \kappa > 0$, for $\delta \in (0,1]$, by setting $k_t = \lceil \left(C_2 T \epsilon^2/b\right)^{\frac{1}{\kappa+1}} \rceil$ and $m_T = \frac{T}{2k_T}$, by lemma 14 in Antos et al. (2008), we have

$$C_1\left(\frac{1}{\epsilon}\right)^D \exp\left(-4C_2 m_T \epsilon^2\right) + 2m_T \beta_{k_T} < \delta,$$

with $\epsilon = \sqrt{\frac{M(\max(M/b,1))^{1/\kappa}}{C_2 t}}$ where $M = \frac{D}{2}\log T + \log(e/\delta) + \log^+ 2\left(\max\left(C_1 C_2^{D/2}, \bar{\beta}\right)\right)$. ■

With the statistical error bound provided in Theorem 6 for solving the derived saddle point problem with arbitrary learnable nonlinear approximators using off-policy samples, we can achieve the analysis of the total error, *i.e.*,

Theorem 7 *Let $\hat{V}_w^T$ be a candidate solution output from the proposed algorithm based on off-policy samples, with at least probability $1-\delta$, we have*

$$\begin{aligned}
\left\|\widehat{V}_w^N - V^*\right\|_{\mu\pi_b}^2 \leqslant\ & \underbrace{2\left(6(K+C_\infty)\epsilon_{app}^\nu + C_\nu(1+\gamma)\epsilon_{app}^V(\lambda) + 3C_\nu\epsilon_{app}^\pi(\lambda)\right)}_{\text{approximation error due to parametrization}} \\
& + \underbrace{16\lambda^2 C_\pi^2 + \left(2\gamma^2+2\right)\left(\frac{\gamma\lambda}{1-\gamma}\max_{\pi\in\mathcal{P}} H(\pi)\right)^2}_{\text{bias due to smoothing}} + \underbrace{4\sqrt{\frac{M\left(\max\left(M/b,1\right)\right)^{1/\kappa}}{C_2 T}}}_{\text{statistical error}} + \underbrace{\left\|\widehat{V}_w^N - \widehat{V}_w^*\right\|_{\mu\pi_b}^2}_{\text{optimization error}}.
\end{aligned}$$

where M is defined as above.

This theorem can be proved by combining Theorem 6 into Lemma 12.

D.3. Convergence Analysis

As we discussed in Section 5.1, the SBEED algorithm converges to a stationary point if we can achieve the optimal solution to the dual functions. However, in general, such conditions restrict the parametrization of the dual functions. In this section, we first provide the proof for Theorem 5. Then, we provide a variant of the SBEED in Algorithm 2, which still achieve the asymptotic convergence with arbitrary function approximation for the dual function, including neural networks with smooth activation functions.

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 Theorem 5 straightforwardly generalizes the convergence result in Ghadimi & Lan (2013) to saddle-point optimization.

Algorithm 2 A variant of SBEED learning

Initialize $w = (w_V, w_\pi, w_\rho)$ and π_b randomly, set ϵ.
for episode $i = 1, \ldots, T$ **do**
  for size $k = 1, \ldots, K$ **do**
    Add new transition (s, a, r, s') into $\mathcal{D}$ by executing behavior policy π_b.
  end for
  for iteration $j = 1, \ldots, N$ **do**
    Sample mini-batch $\{s, a, s'\}^m \sim \mathcal{D}$.
    Compute the stochastic gradient w.r.t. w_ρ as $G_\rho = -\frac{1}{m}\sum_{\{s,a,s'\}\sim\mathcal{D}} \left(\delta(s,a,s') - \rho(s,a)\right) \nabla_{w_\rho}\rho(s,a)$
    Compute the stochastic gradients w.r.t. w_V and w_π as (4) with w_ρ^t, denoted as G_V and G_π, respectively.
    Decay the stepsize ξ_j and ζ_j.
    Update the parameters of primal function by solving the prox-mappings, *i.e.*,

$$\begin{array}{ll} \text{update } \rho: & w_\rho^j = P_{w_\rho^{j-1}}(-\xi_j G_\rho) \\ \text{update } V: & w_V^j = P_{w_V^{j-1}}(\zeta_j G_V) \\ \text{update } \pi: & w_\pi^j = P_{w_\pi^{j-1}}(\zeta_j G_\pi) \end{array}$$

  end for
  Update behavior policy $\pi_b = \pi^N$.
end for

Proof As we discussed, given the empirical off-policy samples, the proposed algorithm can be understood as solving $\min_{V_w,\pi_w} \widehat{\ell}_T(V_w, \pi_w) := \widehat{L}_T(V_w, \pi_w; \nu_w^*)$, where $\nu_w^* = \arg\max_{\nu_w} \widehat{L}_T(V_w, \pi_w; \nu_w)$.

Following the Theorem 2.1 in Ghadimi & Lan (2013), as long as the gradients $\nabla_{V_w}\widehat{\ell}_T(V_w, \pi_w)$ and $\nabla_{\pi_w}\widehat{\ell}_T(V_w, \pi_w)$ are unbiased, under the provided conditions, the finite-step convergence rate can be obtained. The unbiasedness of the gradient estimator is already proved in Theorem 4.

■

Next, we will show that in the setting that off-policy samples are given, under some mild conditions on the neural networks parametrization, the Algorithm 2 will achieve a local Nash equilibrium of the empirical objective asymptotically, *i.e.*, $\left(w_V^+, w_\pi^+, w_\rho^+\right)$, such that

$$\nabla_{w_V, w_\pi}\widehat{L}_\eta\left(w_V^+, w_\pi^+, w_\rho^+\right) = 0, \quad \nabla_{w_\rho}\widehat{L}_\eta\left(w_V^+, w_\pi^+, w_\rho^+\right) = 0.$$

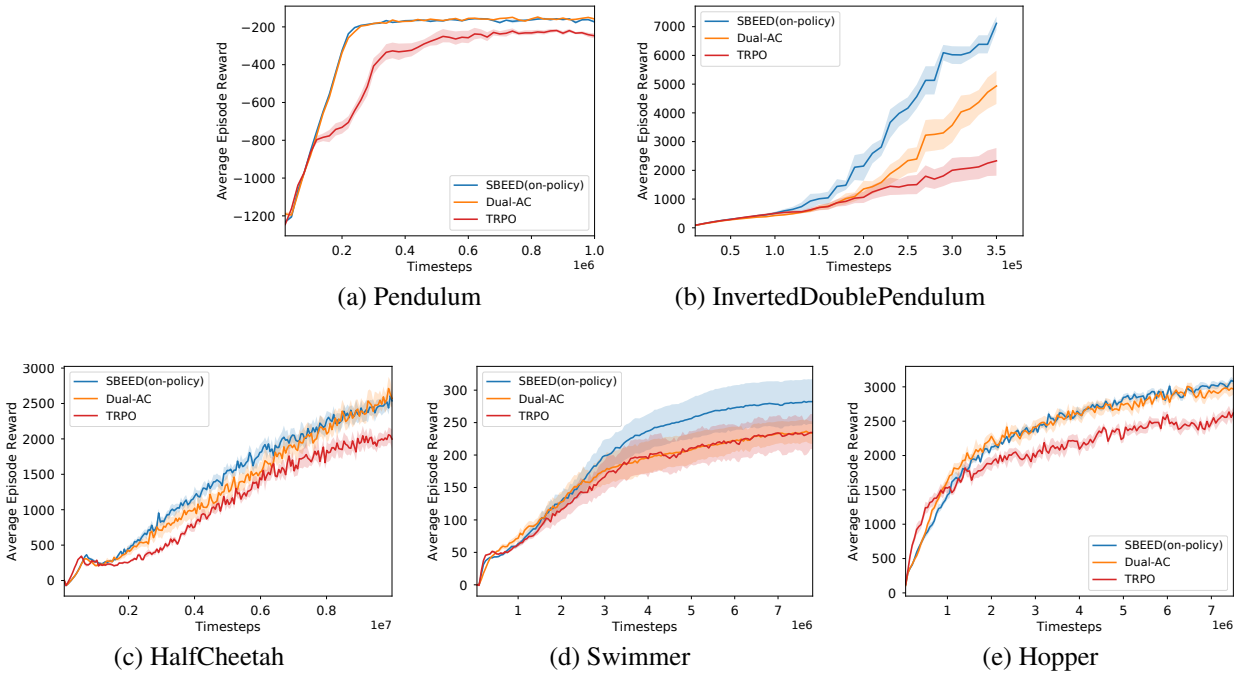
In fact, by applying different decay rate of the stepsizes appropriately for the primal and dual variables in the two time scales updates, the asymptotic convergence of the Algorithm 2 to local Nash equilibrium can be easily obtained by applying the Theorem 1 in Heusel et al. (2017), which is original provided by Borkar (1997). We omit the proof which is not the major contribution of this paper. Please refer to Heusel et al. (2017); Borkar (1997) for further details.

E. More Experiments

E.1. Experimental Details

Policy and value function parametrization The choices of the parametrization of policy are largely based on the recent paper by Rajeswaran et al. (2017), which shows the natural policy gradient with RBF neural network achieves the state-of-the-art performances of TRPO on MuJoCo. For the policy distribution, we parametrize it as $\pi_{\theta_\pi}(a|s) = \mathcal{N}(\mu_{\theta_\pi}(s), \Sigma_{\theta_\pi})$, where $\mu_{\theta_\pi}(s)$ is a two-layer neural nets with the random features of RBF kernel as the hidden layer and the Σ_{θ_π} is a diagonal matrix. The RBF kernel bandwidth is chosen via median trick (Dai et al., 2014; Rajeswaran et al., 2017). Same as Rajeswaran et al. (2017), we use 100 hidden nodes in InvertedDoublePendulum, Swimmer, Hopper, and use 500 hidden nodes in HalfCheetah. This parametrization was used in all on-policy and off-policy algorithms for their policy functions. We adapted the linear parametrization for control variable in TRPO and Dual-AC following Dai et al. (2018). In DDPG and our algorithm SBEED, we need the parametrization for V and ρ (or Q) as fully connected neural networks with two tanh hidden layers with 64 units each.

In the implementation of SBEED, we use the Euclidean distance for w_V and the KL-divergence for w_π in the experiments.



(a) Pendulum (b) InvertedDoublePendulum

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

Figure 3. The results of SBEED against TRPO and Dual-AC in on-policy setting. Each plot shows average reward during training across 5 random runs, with 50% confidence interval. The x-axis is the number of training iterations. SBEED achieves better or comparable performance than TRPO and Dual-AC on all tasks.

We emphasize that other Bregman divergences are also applicable.

Training hyperparameters For all algorithms, we set $\gamma = 0.995$. All V and ρ (or Q) functions of SBEED and DDPG were optimized with ADAM. The learning rates were chosen with a grid search over $\{0.1, 0.01, 0.001, 0.001\}$. For the SBEED, a stepsize of 0.005 was used. For DDPG, an ADAM optimizer was also used to optimize the policy function. The learning rate is set to be $1e-4$ was used. For SBEED, η was set from a grid search of $\{0.004, 0, 01, 0.04, 0.1, 0.04\}$ and λ from a grid search in $\{0.001, 0.01, 0.1\}$. The number of the rollout steps, k was chosen by grid search from $\{1, 10, 20, 100\}$. For off-policy SBEED, a training frequency was chosen from $\{1, 2, 3\} \times 10^3$ steps. A batch size was tuned from $\{10000, 20000, 40000\}$. DDPG updated it's values every iteration and trained with a batch size tuned from $(32, 64, 128)$. For DDPG, τ was set to $1e-3$, reward scaling was set to 1, and the O-U noise σ was set to 0.3.

E.2. On-policy Comparison in Continuous Control Tasks

We compared the SBEED to TRPO and Dual-AC in on-policy setting. We followed the same experimental set up as it is in off-policy setting. We ran the algorithm with 5 random seeds and reported the average rewards with 50% confidence intervals. The empirical comparison results are illustrated in Figure 3. We can see that in all these tasks, the proposed SBEED achieves significantly better performance than the other algorithms. This can be thought as another ablation study that we switch off the "off-policy" in our algorithm. The empirical results demonstrate that the proposed algorithm is more flexible to way of the data sampled.

We set the step size to be 0.01 and the batch size to be 52 trajectories in each iteration in all algorithms in the on-policy setting. For TRPO, the CG damping parameter is set to be 10^{-4}.