A Proof of Theorem 1

Proof. For any trajectory sampled from policy π, if every $s_k, a_k \in \mathcal{SA}_\mu$ then $\sum_{t=0}^{T} \gamma^t r(s_t, a_t) = \sum_{t=0}^{T} \gamma^t r_\mu(s_t, a_t)$. If not, let s_{k+1}, a_{k+1} be the first state-action pair that is not in $\mathcal{SA}_\mu$. Then $\sum_{t=0}^{T} \gamma^t r(s_t, a_t) \geq \sum_{t=0}^{k} \gamma^t r(s_t, a_t) = \sum_{t=0}^{k} \gamma^t r_\mu(s_t, a_t) + \sum_{t=k+1}^{k} \gamma^t r_\mu(s_{abs}, a_t)$. Dividing the accumulated rewards by $\frac{1}{\sum_{t=0}^{T} \gamma^t}$, taking the limit of $T \to \infty$ and expectation over trajectories induced by π, we have that: $R_M^\pi \geq R_{M_\mu}^\pi$. For π^*, since $\mathcal{SA}_\mu$ covers all state-action pairs reachable by π^*, so the expected return remains the same. □

B Proof of Theorem 3

We first state and prove an abstract result. Suppose we have a function $f : \mathbb{R}^d \to \mathbb{R}$ which is differentiable, G-Lipschitz and L-smooth, and f attains a finite minimum value $f^* := \min_{x \in \mathbb{R}^d} f(x)$. Suppose we have access to a noisy gradient oracle which returns a vector $\zeta(x) \in \mathbb{R}^d$ given a query point x. We say that the vector is σ, B-accurate for parameter $\sigma, B \geq 0$ if for all $x \in \mathbb{R}^d$, the quantity $\delta(x) := \zeta(x) - \nabla f(x)$ satisfies

$$\|\mathbb{E}\left[\delta(x) \mid x\right]\| \leq B \quad \text{and} \quad \mathbb{E}\left[\|\delta(x)\|^2 \mid x\right] \leq 2(\sigma^2 + B^2). \tag{5}$$

Notice that the expectations above are only with respect to any randomness in the oracle, while holding the query point fixed. Suppose we run the stochastic gradient descent algorithm using the oracle responses, that is we update $x_{k+1} = x_k - \eta\zeta(x_k)$. While several results for the convergence of stochastic gradient descent to a stationary point of a smooth, non-convex function are well-known, we could not find a result for the biased oracle assumed here and hence we provide a result from first principles. We have the following guarantee on the convergence of the sequence x_k to an approximate stationary point of f.

Theorem 4. *Suppose f is differentiable and L-smooth, and the approximate gradient oracle satisfies the conditions* (5) *with parameters (σ_k, B_k) at iteration k. Then stochastic gradient descent with the oracle, with an initial solution x_1 and stepsize $\eta = 1/L$ satisfies after K iterations:*

$$\frac{1}{K}\sum_{k=1}^{K} \mathbb{E}[\|\nabla f(x_k)\|^2] \leq \frac{2}{K}(f(x_1) - f^*) + \frac{2}{LK}\sum_{k=1}^{K}(\sigma_k^2 + B_k^2).$$

Proof. Since f is L-smooth, we have

$$\begin{aligned}
f(x_{k+1}) &\leq f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k \rangle + \frac{L}{2}\left\|x_{k+1} - x_k\right\|^2 \\
&= f(x_k) - \eta\langle \nabla f(x_k), \zeta(x_k)\rangle + \frac{L\eta^2}{2}\left\|\zeta(x_k)\right\|^2 \\
&= f(x_k) - \eta\langle \nabla f(x_k), \delta(x_k) + \nabla f(x_k)\rangle + \frac{L\eta^2}{2}\left\|\delta(x_k) + \nabla f(x_k)\right\| \\
&= f(x_k) + \left\|\nabla f(x_k)\right\|^2\left(\frac{L\eta^2}{2} - \eta\right) - (\eta - L\eta^2)\langle \nabla f(x_k), \delta(x_k)\rangle + \frac{L\eta^2}{2}\left\|\delta(x_k)\right\|^2.
\end{aligned}$$

Here the first equality follows from our update rule while the remaining simply use the definition of δ along with algebraic manipulations. Now taking expectations of both sides, we obtain

$$\mathbb{E}[f(x_{k+1})] \leq \mathbb{E}[f(x_k)] + \mathbb{E}[\|\nabla f(x_k)\|^2]\left(\frac{L\eta^2}{2} - \eta\right) + (\eta - L\eta^2)GB_k + L\eta^2(\sigma_k^2 + B_k^2),$$

where we have invoked the properties of the oracle to bound the last two terms. Summing over iterations $k = 1, 2, \ldots, K$, we obtain

$$\mathbb{E}[f(x_{k+1})] \leq f(x_1) + \left(\frac{L\eta^2}{2} - \eta\right)\sum_{k=1}^{K}\mathbb{E}[\|\nabla f(x_k)\|^2] + \eta\sum_{k=1}^{K}(GB_k(1 - L\eta) + L\eta(\sigma_k^2 + B_k^2)).$$

Rearranging terms, and using that $f(x_{K+1}) \geq f^*$, we obtain

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}[\|\nabla f(x_k)\|^2] \leq \frac{f(x_1) - f(x^*)}{K(\eta - L\eta^2/2)} + \frac{\eta\sum_{k=1}^{K}(GB_k(1-L\eta) + L\eta(\sigma_k^2 + B_k^2))}{K(\eta - L\eta^2/2)}.$$

Now using the choice $\eta = 1/L$ and simplifying, we obtain the statement of the theorem. □

The theorem tells us that if we pick an iterate uniformly at random from $x_1, \ldots, x_K$, then it is an approximate stationary point in expectation, up to an accuracy which is determined by the bias and variance of the stochastic gradient oracle.

Given this abstract result, we can now prove Theorem 3 by instantiating the errors in the gradient oracle as a function of our assumptions. The proof is similar with convergence proof of stochastic gradient descent from Ghadimi and Lan [2013], while they do not actually cover the case of biased errors in gradients, which is the reason a first-principles proof was included here.

Proof of Theorem 3 We now instantiate the result and assumptions for the case of the off-policy policy gradient method. First, note that the algorithm is stochastic gradient ascent for maximizing the expected return $J(\theta) := R^{\pi_\theta}$. Thus we can apply Theorem 4 with $f = -J$, so that $f(x_1) - f^* \leq V_{\max}$ where $V_{\max}$ is an upper bound on the value of any policy in the MDP. f attains a finite minimum value since the expected return has a finite maximum value. We focus on quantifying the bias B in terms of errors in the critic and propensity score computations first. We first introduce some additional notation. Suppose $w_\theta(s)$ and $Q^\theta(s,a)$ are the true propensity (in terms of state distributions, relative to μ) and Q-value functions for a policy π_θ. Let $g_\theta(s,a) = \frac{\partial \log \pi_\theta(a|s)}{\partial\theta}$. Suppose we are given estimators $\hat{w}$ and $\widehat{Q}$ for w_θ and Q^θ respectively. Then our estimated and true off-policy policy gradients can be written as:

$$\nabla_\theta J(\theta) = \mathbb{E}_\nu[w\rho g_\theta Q^\pi] \quad \text{and} \quad \zeta(\theta) = \hat{w}\rho g_\theta \widehat{Q}.$$

Now the bias can be bounded as

$$\begin{aligned}
\|\mathbb{E}[\zeta(\theta) - \nabla J(\theta)|\theta]\| &= \left\|\mathbb{E}_\nu[w\rho g_\theta Q^\theta - \hat{w}\rho g_\theta\widehat{Q}]\right\| \\
&\leq \left\|\mathbb{E}_\nu[(w-\hat{w})\rho g_\theta Q^\theta]\right\| + \left\|\mathbb{E}_\nu[\hat{w}\rho g_\theta(Q^\theta - \widehat{Q})]\right\| \\
&\leq \mathbb{E}_\nu[\left\|(w-\hat{w})\rho g_\theta Q^\theta\right\|] + \mathbb{E}_\nu[\left\|\hat{w}\rho g_\theta(Q^\theta - \widehat{Q})\right\|] \\
&\leq \mathbb{E}_\nu[|w-\hat{w}|\,\|\rho g_\theta\|\,|Q^\theta|] + \mathbb{E}_\nu[|\hat{w}|\,\|\rho g_\theta\|\,|Q^\theta - \widehat{Q}|]
\end{aligned}$$

By Assumption 2 we have that

$$\|\rho(s,a)g_\theta(s,a)\| = \left\|\frac{1}{\nu(a|s)}\frac{\partial\pi_\theta(a|s)}{\partial\theta}\right\| \leq \frac{G_{\max}}{\nu_{\min}},$$

Then the bound on the bias simplifies to

$$\|\mathbb{E}[\zeta(\theta) - \nabla J(\theta)|\theta]\| \leq \frac{G_{\max}}{\nu_{\min}}\mathbb{E}_\nu[|w-\hat{w}||Q^\theta|] + \frac{G_{\max}}{\nu_{\min}}\mathbb{E}_\nu[|\hat{w}||Q^\theta - \widehat{Q}|]$$

How we simplify further depends on the assumptions we make on the errors in $\hat{w}$ and $\widehat{Q}$. As a natural assumption, suppose that the relative errors are bounded in MSE, that is $\mathbb{E}_\nu(w(s) - \hat{w}(s))^2 \leq \varepsilon_w^2$ and $\mathbb{E}_\nu\left(Q^\theta(s,a) - \widehat{Q}(s,a)\right)^2 \leq \varepsilon_Q^2$. Then by Cauchy-Shwartz inequality, we can simplify the above bias term as

$$\|\mathbb{E}[\zeta(\theta) - \nabla J(\theta)|\theta]\| \leq \frac{G_{\max}}{\nu_{\min}}\varepsilon_w\sqrt{\mathbb{E}_\nu[(Q^\theta)^2]} + \frac{G_{\max}}{\nu_{\min}}\varepsilon_Q\sqrt{\mathbb{E}_\nu[\hat{w}^2]}$$

By Assumption 2 we have $Q^\theta \le V_{\max}$ for all s, a, and

$$\mathbb{E}_\nu[\hat{w}^2] \le \mathbb{E}_\nu[w(s)^2 + (w(s) - \hat{w}(s))^2] \le \mathbb{E}_\nu[w(s)^2] + \mathbb{E}_\nu[(w(s) - \hat{w}(s))^2] \le \sigma_w^2 + \varepsilon_w^2$$

Then the bound on the bias further simplifies to

$$\|\mathbb{E}[\zeta(\theta) - \nabla J(\theta)|\theta]\| \le \varepsilon_w G_{\max} V_{\max}/\nu_{\min} + \varepsilon_Q G_{\max}\sqrt{\sigma_w^2 + \varepsilon_w^2}/\nu_{\min}$$

Similarly, for the variance we have

$$\begin{aligned}\mathbb{E}[\|\zeta(\theta) - \nabla J(\theta)\|^2 |\theta] &\le 2\mathbb{E}_\nu\left[\left\|(\hat{w} - w)\rho g_\theta Q^\theta\right\|^2\right] + 2\mathbb{E}_\nu\left[\left\|\hat{w}\rho g_\theta(\widehat{Q} - Q^\theta)\right\|^2\right] \\ &\le 2\varepsilon_w^2 G_{\max}^2 V_{\max}^2/\nu_{\min}^2 + 2\varepsilon_Q^2(\sigma_w^2 + \varepsilon_w^2)G_{\max}^2/\nu_{\min}^2.\end{aligned}$$

Hence, the RHS of Theorem 4 simplifies to

$$\frac{2V_{\max}}{K} + O\left(\frac{\sum_{k=1}^T \varepsilon_{w,k}^2 G_{\max}^2 V_{\max}^2/\nu_{\min}^2 + \varepsilon_{Q,k}^2(\sigma_w^2 + \varepsilon_{w,k}^2)G_{\max}^2/\nu_{\min}^2}{K}\right),$$

where $\varepsilon_{w,k}$ and $\varepsilon_{Q,k}$ are the error parameters in the propensity scores and critic at the k_{th} iteration of our algorithm. Since we update these quantities online along with the policy parameters, we expect $\varepsilon_{w,k}$ and $\varepsilon_{Q,k}$ to decrease as k increases. That is, assuming that ν satisfies the coverage assumptions with finite upper bounds on the propensities and the policy class is Lipschitz continuous in its parameters, the scheme converges to an approximate stationary point given estimators $\hat{w}$ and $\widehat{Q}$ with a small average MSE across the iterations under ν.

C Details for Hard Example of Off-PAC

Consider the problem instance shown in Figure 1, and the policy class describe in the paper. The true state value function of π_α, V_{π_α} satisfies that: $V^{\pi_\alpha}(s_0) = V^{\pi_\alpha}(s_1) = \frac{1+\alpha}{2},\ V^{\pi_\alpha}(s_2) = \frac{1-\alpha}{2},\ V^{\pi_\alpha}(s_3) = 1,\ V^{\pi_\alpha}(s_4) = 0$. We now study the Off-PAC gradient estimator $g_{\text{OffPAC}}(\alpha)$ in an idealized setting where the critic Q^{π_α} is perfectly known. As per Equation 5 of Degris et al. [2012], we have

$$\begin{aligned}g_{\text{OffPAC}}(\alpha) &= \sum_s d^{\pi_b}(s)\sum_a \frac{\partial \pi_\alpha(a|s)}{\partial \alpha} Q^{\pi_\alpha}(s,a) \\ &= d^{\pi_b}(s_1)\Big(Q^{\pi_\alpha}(s_1,\ell) - Q^{\pi_\alpha}(s_1,r)\Big) \\ &\quad + d^{\pi_b}(s_2)\Big(Q^{\pi_\alpha}(s_2,\ell) - Q^{\pi_\alpha}(s_2,r)\Big) \\ &= \tfrac{1}{2}(1 - 1/2) + \tfrac{1}{2}(0 - 1/2) = 0.\end{aligned}$$

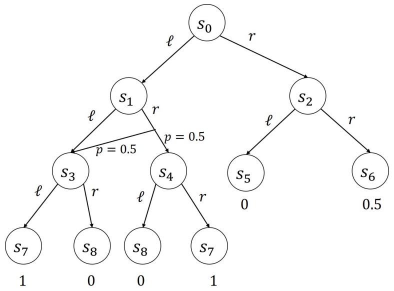
That is, the gradient vanishes for any policy π_α, meaning that the algorithm can be arbitrarily sub-optimal at any point during policy optimization. We note that this does not contradict the previous Off-PAC theorems as the policy class is not fully expressive in our example, a requirement for their convergence results.

D Details for Experiments

In this section we will show some important details and hyper-parameter settings of our algorithm in experiments.

D.1 Simulation Domains

For the first domain *cart pole* control problem, the state space is continuous and describes the position and velocity of cart and pole. The action space consists of applying a unit force to two directions. The horizon is fixed to 200. If the



trajectory ends in less than 200 steps, we pad the episode by continuing to sample actions and repeating the last state. We use a uniformly random policy to collect $n = 500$ trajectories as off-policy data. We use neural networks with a 32-unit hidden layer to fit the stationary distribution ratio, actor and critic.

The second domain is an *HIV treatment simulation* described in Ernst et al. [2006]. The transition dynamics are modeled by an ODE system in Ernst et al. [2006]. The reward consists of a small negative reward for deploying each type of drug, and a positive reward based on the HIV-specific cytotoxic T-cells which will increase with a proper treatment schedule. To maximize the total reward in this simulator, algorithms need to do structured treatment interruption (STI), which aim to achieve a balance between treatment and the adverse effect of abusing drugs. Each trajectory simulates a treatment period for one patient in 1000 days and each step corresponds to a 5-day interval in the ODE system. We represent the state by taking logarithm of state features and divide the reward by 10^8 to ensure they are in a reasonable range to fit the neural network models. We use neural networks with three 16-unit hidden layers to fit the actor and state distribution ratio, and a neural network with four 32-unit hidden layers for the critic.

D.2 Hyper-parameters

We use three separate neural networks, one for each of actor, critic, and the state distribution ratio model w. We use the Adam optimizer for all of them. We also use a entropy regularization for the actor. We warm start the actor by maximizing the log-likelihood of actor on the collected dataset. For critic, we use the same critic algorithm as we used in Algorithm 1 except that there is no importance sampling ratio (as it is on-policy for the warm start). For the warm start of w, we just fit the w for several iterations using the warm start policy found for the actor. Warm start uses the same learning rates as normal training. For critic and w, we also keep the state of optimizer to be the same when we start normal training.

Hyper-parameters	cart pole	HIV
γ	1.0	0.98
λ	0.	0.
entropy coefficient	0.01	0.03
learning rate (actor)	1e-3	5e-6
learning rate (critic)	1e-3	1e-3
learning rate (w)	1e-3	3e-4
batch size (actor)	5000	5000
batch size (critic)	5000	5000
batch size (w)	200	200
number of iterations (critic)	10	10
number of iterations (w)	50	50
weight decay (w)	1e-5	1e-5
behavior cloning number of iterations	2000	2000
warm start number of iterations (crtic)	500	2500
warm start number of iterations (w)	500	2500

Table 1: Hyper-parameters in experiments

In the Table 1 we show some hyper-parameters setting we used in both domain. We also follow the details in Algorithm

1 and Algorithm 2 of Liu et al. [2018a] to learn w. We scale the inputs to w so that the whole off-policy dataset has zero mean and standard deviation of 1 along each dimension in state space. We use the RBF kernel to compute the loss function for w. For the cart pole simulator, the kernel bandwidth is set to be the median of state distance. If computing this median state distance over the whole off-policy dataset is computationally too expensive, it can be approximated using a mini-batch. In the HIV domain the bandwidth is set to be 1. When we compute the loss of w, we need to sample two mini-batch independently to get an unbiased estimates of the quadratic loss. The loss in each pair of mini-batch is normalized by the sum of kernel matrix elements computed from them.

D.3 Choice of Algorithm with Discounted Reward

In discounted reward settings, the state distribution is also defined with respect to the discount factor γ, and Liu et al. [2018a] introduce an algorithm to learn state distribution ratio in this setting. However, we notice that in on-policy policy learning cases, though the policy gradient theorem [Sutton et al., 2000] requires samples from the stationary state distribution defined using the discount factor, it is common to directly use the collected samples to compute policy gradient without re-sampling/re-weighting (s, a, r, s')'s according to the discounted stationary distribution. This might be driven by sample efficiency concerns, as samples at later time-steps in the discounted stationary distribution will receive exponentially small probability, meaning they are not leveraged as effectively by the algorithm. Given this, we compare three different variants of our algorithm in HIV experiment with discounted reward. The first (OPPOSD average) variant uses the algorithm for the average reward setting, but evaluates its discounted reward. The second learns the state distribution ratio $w(s)$ in the discounted case (Algorithm 2 in [Liu et al., 2018a]), but still samples from the undiscounted distribution to compute the gradient (OPPOSD disc w). The third learns the state distribution ratio $w(s)$ in the discounted case and also re-samples the samples according to $d^\pi(s) = \lim_{T\to\infty} \frac{1}{\sum_{t=0}^{T}\gamma^t} \sum_{t=0}^{T} \gamma^t d_t^\pi(s)$ (OPPOSD). In the main body of paper, we select the third one as it is the most natural way from the definition of problem and policy gradient theorem. Results of these three methods are demonstrated in Figure 4 and they do not have significant differences in this experiment.

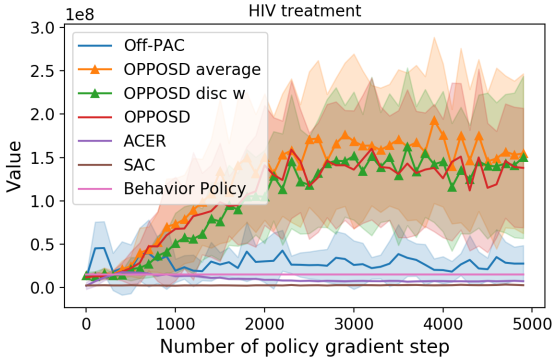


Figure 4: Episodic scores over length 200 episodes in HIV treatment simulator.

E Discussion on Related Off-Policy Learning Algorithms

There are also many different algorithms which have been built using Off-PAC [Degris et al., 2012] and improve Off-PAC in different directions, such as DDPG [Lillicrap et al., 2015], ACER [Wang et al., 2016], etc. They are orthogonal to our work and our state distribution correction techniques are composable with these further improvements in the Off-PAC framework. For understanding the impact of correcting the stationary distribution, in the experiment section of this work we therefore focus on ablation comparison with Off-PAC. It would be interesting to combine our work with the additional contributions of DDPG, ACER etc. to derive improved variants of each of those algorithms.

We also wish to clarify that some previous off-policy policy gradient algorithms such as DDPG, SAC [Haarnoja et al., 2018] and ACER focus on a different setting with this paper – they consider online off policy where data is collected every iteration using the current policy with potentially noise, and the off policy nature comes from when updating to a new policy, the algorithms use all the data collected across previous iterations. In contrast our focus and experiments are on batch off policy learning, where data is collected in advance from a behavior policy and a new policy is computed using only that batch dataset. The difference between the online off-policy and batch off-policy settings is important since algorithms that receive periodic access to new samples gathered using the current policy may benefit significantly. To illustrate the difference in the two settings, we ran SAC and related ACER in our batch off-policy setting in the experimental section. SAC is proposed for continuous actions setting in their paper and available code, and we re-derive the policy gradient updates for discrete actions.

We separate out discussion of SBEED [Dai et al., 2018] since its theoretical results are derived for a similar batch off policy RL setting as our approach. However, it has not been experimentally evaluated in a batch off-policy setting (their empirical results were for off policy RL as described above where more data is collected). SBEED advances over a number of prior theoretical results on sample complexity for value function approaches for batch off policy learning [Antos et al., 2008]. In contrast, in policy gradient methods there has not even existed a statistically consistent procedure for batch off-policy learning, and this is the fundamental contribution of our work. In many domains policy optimization techniques are the methods of choice, and policy improvement from a reasonable policy is often more natural in safety critical applications. Since these are the types of applications where off-policy carries the most appeal, we focus on the class of policy optimization algorithms that can work in a batch off-policy setting. We can possibly leverage similar techniques in value function learning too, and we view that development along with a detailed evaluation against other value function learning methods as future work.