A Proof of Theorem 3.1

Proof of Theorem 3.1. To prove Eq. 3 for Alg. 1, we use the proof techniques from Flaxman et al. (2005). The proof is more simpler than the one in Flaxman et al. (2005) as we do not have to deal with shrinking and reshaping the predictor set Θ.

Denote $u \sim \mathbb{B}_b$ as uniformly sampling u from a b-dim unit ball, $u \sim \mathbb{S}_b$ as uniformly sampling u from the b-dim unit sphere, and $\delta \in (0,1)$. Consider the loss function $\hat{c}_i(w_i) = \mathbb{E}_{v\sim\mathbb{B}_b}[c_i(\theta_i + \delta v)]$, which is a smoothed version of $c_i(w_i)$. It is shown in Flaxman et al. (2005) that the gradient of $\hat{c}_i$ with respect to θ is:

$$\begin{aligned}&\nabla_\theta \hat{c}_i(\theta)|_{\theta=\theta_i}\\ &= \frac{b}{\delta}\mathbb{E}_{u\sim\mathbb{S}_b}[c_i(\theta_i+\delta u)u]\\ &= \frac{b}{\delta}\mathbb{E}_{u\sim\mathbb{S}_b}[((\theta_i+\delta u)^T s_i - a_i)^2 u].\end{aligned}$$

Hence, the descent direction we take in Alg. 1 is actually an unbiased estimate of $\nabla_\theta \hat{c}_i(\theta)|_{\theta=\theta_i}$. So Alg. 1 can be considered as running OGD with an unbiased estimate of gradient on the sequence of loss $\hat{c}_i(\theta_i)$. It is not hard to show that for an unbiased estimate of $\nabla_\theta \hat{c}_i(\theta)|_{\theta=\theta_i} = \frac{b}{\delta}((\theta_i+\delta u)^T s_i - a_i)^2 u$, the norm is bounded as $b(C^2+C_s^2)/\delta$. Now we can directly applying Lemma 3.1 from Flaxman et al. (2005), to get:

$$\mathbb{E}\left[\sum_{i=1}^T \hat{c}_i(\theta_i)\right] - \min_{\theta^\star\in\Theta}\sum_{i=1}^T \hat{c}_i(\theta^\star) \leq \frac{C_\theta b(C^2+C_s^2)}{\delta}\sqrt{T}. \tag{8}$$

We can bound the difference between $\hat{c}_i(\theta)$ and $c_i(\theta)$ using the Lipschitiz continuous property of c_i:

$$\begin{aligned}|\hat{c}_i(\theta) - c_i(\theta)| &= |\mathbb{E}_{v\sim\mathbb{B}_b}[c_i(\theta+\delta v) - c_i(\theta)]|\\ &\leq \mathbb{E}_{v\sim\mathbb{B}_b}[|c_i(\theta+\delta v) - c_i(\theta)|] \leq L\delta.\end{aligned} \tag{9}$$

Substitute the above inequality back to Eq. 8, rearrange terms, we get:

$$\begin{aligned}&\mathbb{E}\left[\sum_{i=1}^T c_i(\theta_i)\right] - \min_{\theta^\star\in\Theta}\sum_{i=1}^T c_i(w^\star)\\ &\leq \frac{C_\theta b(C^2+C_s^2)}{\delta}\sqrt{T} + 2LT\delta.\end{aligned} \tag{10}$$

By setting $\delta = T^{-0.25}\sqrt{\frac{C_\theta b(C^2+C_s^2)}{2L}}$, we get:

$$\begin{aligned}&\mathbb{E}\left[\sum_{i=1}^T c_i(\theta_i)\right] - \min_{w^\star\in\Theta}\sum_{i=1}^T c_i(w^\star)\\ &\leq \sqrt{C_\theta b(C^2+C_s^2)L}T^{3/4}.\end{aligned}$$

To prove Eq. 4 for Alg. 4, we follow the similar strategy in the proof of Alg. 1.

Denote $\epsilon \sim [-1,1]$ as uniformly sampling ϵ from the interval $[-1,1]$, $e \sim \{-1,1\}$ as uniformly sampling e from the set containing -1 and 1. Consider the loss function $\tilde{c}_i(\theta) = \mathbb{E}_{\epsilon\sim[-1,1]}[(\theta^T s_i + \delta\epsilon - a_i)^2]$. One can show that the gradient of $\tilde{c}_i(\theta)$ with respect to θ is:

$$\nabla_\theta \tilde{c}_i(\theta) = \frac{1}{\delta}\mathbb{E}_{e\sim\{-1,1\}}[e(\theta^\top s_i + \delta e - a_i)^2 s_i]. \tag{11}$$

As we can see that the descent direction we take in Alg. 4 is actually an unbiased estimate of $\nabla_\theta \tilde{c}_i(\theta)|_{\theta=\theta_i}$. Hence Alg. 4 can be considered as running OGD with unbiased estimates of gradients on the sequence of loss functions $\tilde{c}_i(\theta)$. For an unbiased estimate of the gradient, $\frac{1}{\delta}e(\theta_i^\top s_i + \delta e - a_i)^2 s_i$, its norm is bounded as $(C^2+1)C_s/\delta$. Note that different from Alg. 1, here the maximum norm of the unbiased gradient *is independent of feature dimension* b. Now we apply Lemma 3.1 from Flaxman et al. (2005) on $\tilde{c}_i$, to get:

$$\mathbb{E}\left[\sum_{i=1}^{T}\tilde{c}_i(\theta_i)\right] - \min_{\theta^*\in\Theta}\sum_{i=1}^{T}\tilde{c}_i(\theta^*) \leq \frac{C_\theta(C^2+1)C_s}{\delta}\sqrt{T}. \tag{12}$$

Again we can bound the difference between $\tilde{c}_i(\theta)$ and $c_i(\theta)$ for any θ using the fact that $(\hat{a}_i - a_i)^2$ is Lipschitz continuous with respect to prediction $\hat{a}_i$ with Lipschitz constant C:

$$\begin{aligned}|\tilde{c}_i(\theta) - c_i(\theta)| &= |\mathbb{E}_{\epsilon\sim[-1,1]}[(\theta^\top s_i + \delta\epsilon - a_i)^2 - (\theta^\top s_i - a_i)^2]| \\ &\leq \mathbb{E}_{\epsilon\sim[-1,-1]}[C\delta|\epsilon|] \leq C\delta. \end{aligned} \tag{13}$$

Substitute the above inequality back to Eq. 12, rearrange terms:

$$\begin{aligned}&\mathbb{E}\left[\sum_{i=1}^{T}\tilde{c}_i(\theta_i)\right] - \min_{\theta^*\in\Theta}\sum_{i=1}^{T}\tilde{c}_i(\theta^*) \\ &\leq \frac{C_\theta(C^2+1)C_s}{\delta}\sqrt{T} + 2C\delta T.\end{aligned}$$

Set $\delta = T^{-0.25}\sqrt{\frac{C_\theta(C^2+1)C_s}{2C}}$, we get:

$$\begin{aligned}&\mathbb{E}\left[\sum_{i=1}^{T}\tilde{c}_i(\theta_i)\right] - \min_{\theta^*\in\Theta}\sum_{i=1}^{T}\tilde{c}_i(\theta^*) \\ &\leq \sqrt{C_\theta(C^2+1)C_sC}T^{3/4}.\end{aligned}$$

□

B Proof of Theorem 4.1

We first present some useful lemmas below.

Consider the smoothed objective given by $\hat{J}(\theta) = \mathbb{E}_{v\sim\mathbb{B}_d}[J(\theta+\delta v)]$ where $\mathbb{B}_d$ is the unit ball in d dimensions and δ is a positive constant. Using the assumptions stated in Section 4.1, we obtain the following useful lemma:

Lemma B.1. *If the objective $J(\theta)$ satisfies the assumptions in Section 4.1 and the smoothed objective $\hat{J}(\theta)$ is as given above, then we have that*

1. *$\hat{J}(\theta)$ is also G-Lipschitz and L-smooth*
2. *For all $\theta \in \mathbb{R}^d$, $\|\nabla_\theta J(\theta) - \nabla_\theta \hat{J}(\theta)\| \leq L\delta$*

Proof of Lemma B.1. Consider for any $\theta_1, \theta_2 \in \mathbb{R}^d$,

$$\begin{aligned}|\hat{J}(\theta_1) - \hat{J}(\theta_2)| &= |\mathbb{E}_{v\sim\mathbb{B}_d}[J(\theta_1+\delta v) - J(\theta_2+\delta v)]| \\ &\leq \mathbb{E}_{v\sim\mathbb{B}_d}[|J(\theta_1+\delta v) - J(\theta_2+\delta v)|] \\ &\leq \mathbb{E}_{v\sim\mathbb{B}_d}[G\|\theta_1-\theta_2\|] \\ &= G\|\theta_1-\theta_2\|\end{aligned}$$

The above inequalities are due to the fact that expectation of absolute value is greater than absolute value of expectation, and the G-lipschitz assumption on $J(\theta)$. Thus, the smoothened loss function $\hat{J}(\theta)$ is also G-lipschitz. Similarly consider,

$$\begin{aligned}&\|\nabla_\theta\hat{J}(\theta_1) - \nabla_\theta\hat{J}(\theta_2)\| \\ &\quad = \|\nabla_\theta\mathbb{E}_{v\sim\mathbb{B}_d}[J(\theta_1+\delta v)] - \nabla_\theta\mathbb{E}_{v\sim\mathbb{B}_d}[J(\theta_2+\delta v)]\|\end{aligned}$$

$$
\begin{aligned}
&= \|\mathbb{E}_{v\sim\mathbb{B}_d}[\nabla_\theta J(\theta_1+\delta v) - \nabla_\theta J(\theta_2+\delta v)]\| \\
&\leq \mathbb{E}_{v\sim\mathbb{B}_d}[\|\nabla_\theta J(\theta_1+\delta v) - \nabla_\theta J(\theta_2+\delta v)\|] \\
&\leq \mathbb{E}_{v\sim\mathbb{B}_d}[L\|\theta_1-\theta_2\|] \\
&= L\|\theta_1-\theta_2\|
\end{aligned}
$$

The above inequalities are due to the fact that expectation of norm is greater than norm of expectation, and the L-smoothness assumption on $J(\theta_1)$. We interchange the expectation and derivative using the assumptions on $J(\theta_1)$ and the dominated convergence theorem. Thus, the smoothened loss function $\hat{J}(\theta_1)$ is also L-smooth.

We know,

$$
\begin{aligned}
\nabla_\theta \hat{J}(\theta) &= \nabla_\theta \mathbb{E}_{v\sim\mathbb{B}_d}[J(\theta+\delta v)] \\
&= \mathbb{E}_{v\sim\mathbb{B}_d}[\nabla_\theta J(\theta+\delta v)]
\end{aligned}
$$

Note that the expectation and derivative can be interchanged using the dominated convergence theorem. Hence, we have

$$
\begin{aligned}
\|\nabla_\theta \hat{J}(\theta) - \nabla_\theta J(\theta)\| &= \|\mathbb{E}_{u\sim\mathbb{B}_d}[\nabla_\theta J(\theta+\delta v)] - \nabla_\theta J(\theta)\| \\
&\leq \mathbb{E}_{u\sim\mathbb{B}_d}\|\nabla_\theta J(\theta+\delta v) - \nabla_\theta J(\theta)\| \\
&\leq \mathbb{E}_{u\sim\mathbb{B}_d}[L||\delta v||] \\
&\leq L\delta
\end{aligned}
$$

□

The above lemma will be very useful later when we try to relate the convergence rate for the smoothed objective and the true objective. It is shown in (Flaxman et al., 2005; Agarwal et al., 2010) that the gradient estimate g_i is an unbiased estimator of the gradient $\nabla_\theta \hat{J}(\theta_i)$. Hence, Algorithm 3 is performing SGD on the smoothed objective $\hat{J}(\theta)$. Using this insight, we can use the convergence rate of SGD for nonconvex functions to stationary points from (Ghadimi and Lan, 2013) which is given as follows

Lemma B.2 ((Ghadimi and Lan, 2013))**.** *Consider running SGD on the objective $\hat{J}(\theta)$ that is L-smooth and G-Lipschitz for T steps. Fix initial solution θ_0 and denote $\Delta_0 = \hat{J}(\theta_0) - \hat{J}(\theta^*)$ where θ^* is the point at which $\hat{J}(\theta)$ attains global minimum. Also, assume that the gradient estimate g_i is unbiased and has a bounded variance, i.e. for all i, $\mathbb{E}_i[\|g_i - \nabla_\theta \hat{J}(\theta_i)\|_2^2] \leq V \in \mathbb{R}^+$ where $\mathbb{E}_i$ denotes expectation with randomness only at iteration i conditioned on history upto iteration $i-1$. Then we have,*

$$
\frac{1}{T}\sum_{i=1}^{T} \mathbb{E}\|\nabla_\theta \hat{J}(\theta_i)\|_2^2 \leq \frac{2\sqrt{2\Delta_0 L(V+G^2)}}{\sqrt{T}} \tag{14}
$$

For completeness, we include a proof of the above lemma below.

Proof of Lemma B.2. Denote $\xi_i = g_i - \nabla_\theta \hat{J}(\theta_i)$. Note that $\mathbb{E}_i[\xi_i] = 0$ since the stochastic gradient g_i is unbiased. From $\theta_{i+1} = \theta_i - \alpha g_i$, we have:

$$
\begin{aligned}
\hat{J}(\theta_{i+1}) &= \hat{J}(\theta_i - \alpha g_i) \\
&\leq \hat{J}(\theta_i) - \nabla_\theta \hat{J}(\theta_i)^\top(\alpha g_i) + \frac{L\alpha^2}{2}\|g_i\|_2^2 \\
&= \hat{J}(\theta_i) - \alpha\nabla_\theta \hat{J}(\theta_i)^\top g_i + \frac{L\alpha^2}{2}\|\xi_i + \nabla_\theta \hat{J}(\theta_i)\|_2^2 \\
&= \hat{J}(\theta_i) - \alpha\nabla_\theta \hat{J}(\theta_i)^\top g_i + \frac{L\alpha^2}{2}(\|\xi_i\|_2^2 \\
&\quad + 2\xi_i^\top \nabla_\theta \hat{J}(\theta_i) + \|\nabla_\theta \hat{J}(\theta_i)\|_2^2)
\end{aligned}
$$

The first inequality above is obtained since the loss function $\hat{J}(\theta)$ is L-smooth. Adding $\mathbb{E}_i$ on both sides and using the fact that $\mathbb{E}_i[\xi_i] = 0$, we have:

$$
\mathbb{E}_i[\hat{J}(\theta_{i+1})] = \hat{J}(\theta_i) - \alpha\|\nabla_\theta \hat{J}(\theta_i)\|_2^2
$$

$$
\begin{aligned}
&+ \frac{L\alpha^2}{2}\left(\mathbb{E}_i[\|\xi_i\|_2^2] + \|\nabla_\theta \hat{J}(\theta_i)\|_2^2\right) \\
&\leq \hat{J}(\theta_i) - \alpha\|\nabla_\theta \hat{J}(\theta_i)\|_2^2 \\
&+ \frac{L\alpha^2}{2}\left(\mathbb{E}_i[\|\xi_i\|_2^2] + G^2\right)
\end{aligned}
$$

where the inequality is due to the lipschitz assumption. Rearranging terms, we get:

$$
\begin{aligned}
\alpha\|\nabla_\theta \hat{J}(\theta_i)\|_2^2 &= \hat{J}(\theta_i) - \mathbb{E}_i[\hat{J}(\theta_{i+1})] \\
&+ \frac{L\alpha^2}{2}(\mathbb{E}_i[\|\xi_i\|_2^2] + G^2) \\
&\leq \hat{J}(\theta_i) - \mathbb{E}_i[\hat{J}(\theta_{i+1})] + \frac{L\alpha^2}{2}(V + G^2)
\end{aligned}
$$

Sum over from time step 1 to T, we get:

$$
\begin{aligned}
\alpha\sum_{t=1}^{T}\mathbb{E}\|\nabla_\theta \hat{J}(\theta_i)\|_2^2 &\leq \mathbb{E}[\hat{J}(\theta_0) - \hat{J}(\theta_T)] \\
&+ \frac{LT\alpha^2}{2}(V + G^2)
\end{aligned}
$$

Divide α on both sides, we get:

$$
\begin{aligned}
\sum_{t=1}^{T}\mathbb{E}\|\nabla_\theta \hat{J}(\theta_i)\|_2^2 &\leq \frac{1}{\alpha}\mathbb{E}[\hat{J}(\theta_0) - \hat{J}(\theta_T)] + LT\alpha(V + G^2) \\
&\leq \frac{1}{\alpha}\mathbb{E}[\hat{J}(\theta_0) - \hat{J}(\theta^*)] + LT\alpha(V + G^2) \\
&= \frac{1}{\alpha}\Delta_0 + LT\alpha(V + G^2) \\
&\leq \sqrt{\frac{\Delta_0 LT(V + G^2)}{2}} + \sqrt{2\Delta_0 LT(V + G^2)} \\
&\leq 2\sqrt{2\Delta_0 LT(V + G^2)}
\end{aligned}
$$

with $\alpha = \sqrt{\frac{2\Delta_0}{LT(V+G^2)}}$. Hence, we have:

$$
\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla_\theta \hat{J}(\theta_i)\|_2^2 \leq \frac{2\sqrt{2\Delta_0 L(V + G^2)}}{\sqrt{T}}
$$

□

The above lemma is useful as it gives us the following result:

$$
\begin{aligned}
\min_{1\leq i\leq T}\mathbb{E}\|\nabla_\theta \hat{J}(\theta_i)\|_2^2 &\leq \frac{1}{T}\sum_{i=1}^{T}\mathbb{E}\|\nabla_\theta \hat{J}(\theta_i)\|_2^2 \\
&\leq \frac{2\sqrt{2\Delta_0 L(V + G^2)}}{\sqrt{T}}
\end{aligned}
\tag{15}
$$

since the minimum is always less than the average. We have then that using SGD to minimize a nonconvex objective finds a θ_i that is 'almost' a stationary point in bounded number of steps provided the stochastic gradient estimate has bounded variance.

We now show that the gradient estimate g_i used in Algorithm 3 indeed has a bounded variance. Observe that the estimate g_i in the algorithm is a two-point estimate, which should have substantially less variance than one-point estimates (Agarwal et al., 2010). However, the two evaluations, resulting in J_i^+ and J_i^-, have different independent noise. This is due to the fact that in policy search, stochasticity arises from the environment and cannot be controlled and we cannot obtain the significant variance reduction that is typical of two-point estimators. The following lemma quantifies the bound on the variance of gradient estimate g_i:

Lemma B.3. *Consider a smoothed objective $\hat{J}(\theta) = \mathbb{E}_{v\sim\mathbb{B}_d}[J(\theta + \delta v)]$ where $\mathbb{B}_d$ is the unit ball in d dimensions, $\delta > 0$ is a scalar and the true objective $J(\theta)$ is G-lipschitz. Given gradient estimate $g_i = \frac{d(J_i^+ - J_i^-)}{2\delta}u$ where u is sampled uniformly from a unit sphere $\mathbb{S}_d$ in d dimensions, $J_i^+ = J(\theta_i + \delta u) + \eta_i^+$ and $J_i^- = J(\theta - \delta u) + \eta_i^-$ for zero mean random i.i.d noises η_i^+, η_i^-, we have*

$$\mathbb{E}_i[\|g_i - \nabla_\theta \hat{J}(\theta_i)\|_2^2] \leq 2d^2G^2 + 2\frac{d^2\sigma^2}{\delta^2} \tag{16}$$

where σ^2 is the variance of the random noise η.

Proof of Lemma B.3. From Shamir (2017), we know that g_i is an unbiased estimate of the gradient of $\hat{J}(\theta_i)$, i.e. $\mathbb{E}_{u_i\sim\mathbb{S}_d}[g_i] = \nabla\hat{J}(\theta_i)$. Thus, we have

$$\begin{aligned}
&\mathbb{E}_{u_i\sim\mathbb{S}_d}\|g_i - \nabla\hat{J}(\theta_i)\|^2 \\
&\quad= \mathbb{E}_{u_i\sim\mathbb{S}_d}[\|g_i\|^2 + \|\nabla\hat{J}(\theta)_i\|^2 - 2g_i^T\nabla\hat{J}(\theta_i)] \\
&\quad= \mathbb{E}_{u_i\sim\mathbb{S}_d}\|g_i\|^2 + \|\nabla\hat{J}(\theta_i)\|^2 - 2\|\nabla\hat{J}(\theta_i)\|^2 \\
&\quad= \mathbb{E}_{u_i\sim\mathbb{S}_d}\|g_i\|^2 - \|\nabla\hat{J}(\theta_i)\|^2 \\
&\quad\leq \mathbb{E}_{u_i\sim\mathbb{S}_d}\|g_i\|^2 \\
&\quad= \frac{d^2}{4\delta^2}\mathbb{E}_{u_i\sim\mathbb{S}_d}\|(J(\theta_i + \delta u_i) - J(\theta_i - \delta u_i) \\
&\qquad+ (\eta_i^+ - \eta_i^-))u_i\|^2 \\
&\quad\leq \frac{d^2}{2\delta^2}[\mathbb{E}_{u_i\sim\mathbb{S}_d}\|(J(\theta_i + \delta u_i) - J(\theta_i - \delta u_i)u_i\|_2^2 \\
&\qquad+ \mathbb{E}_{u_i\sim\mathbb{S}_d}\|(\eta_i^+ - \eta_i^-))u_i\|^2] \\
&\quad\leq \frac{d^2}{2\delta^2}[\mathbb{E}_{u_i\sim\mathbb{S}_d}4G^2\delta^2\|u_i\|^2 + 4\mathbb{E}_{u_i\sim\mathbb{S}_d}\|\eta_i^+\|_2^2\|u_i\|_2^2] \\
&\quad= 2d^2G^2 + 2\frac{d^2\sigma^2}{\delta^2}
\end{aligned}$$

where the second inequality is true as $\|a + b\|_2^2 \leq 2(\|a\|_2^2 + \|b\|_2^2)$ and the last inequality is due to the Lipschitz assumption on $J(\theta)$. □

We are ready to prove Theorem 4.1.

Proof of Theorem 4.1. Fix initial solution θ_0 and denote $\Delta_0 = \hat{J}(\theta_0) - \hat{J}(\theta^*)$ where $\hat{J}(\theta)$ is the smoothed objective and θ^* is the point at which $\hat{J}(\theta)$ attains global minimum. Since the gradient estimate g_i used in Algorithm 3 is an unbiased estimate of the gradient $\nabla_\theta\hat{J}(\theta_i)$, we know that Algorithm 3 performs SGD on the smoothed objective. Moreover, from Lemma B.3, we know that the variance of the gradient estimate g_i is bounded. Hence, we can use Lemma B.2 on the smoothed objective $\hat{J}(\theta)$ to get

$$\frac{1}{T}\sum_{i=1}^{T}\mathbb{E}\|\nabla_\theta\hat{J}(\theta_i)\|_2^2 \leq \frac{2\sqrt{2\Delta_0 L(V + G^2)}}{\sqrt{T}} \tag{17}$$

where $V \leq 2d^2G^2 + 2\frac{d^2\sigma^2}{\delta^2}$ (from Lemma B.3). We can relate $\nabla_\theta\hat{J}(\theta)$ and $\nabla_\theta J(\theta)$ - the quantity that we ultimately care about, as follows:

$$\begin{aligned}
&\frac{1}{T}\sum_{i=1}^{T}\mathbb{E}\|\nabla_\theta J(\theta_i)\|_2^2 \\
&\quad= \frac{1}{T}\sum_{i=1}^{T}\mathbb{E}\|\nabla_\theta J(\theta_i) - \nabla_\theta\hat{J}(\theta_i) + \nabla_\theta\hat{J}(\theta_i)\|_2^2
\end{aligned}$$

$$\leq \frac{2}{T}\sum_{i=1}^{T} \mathbb{E}\|\nabla_\theta J(\theta_i) - \nabla_\theta \hat{J}(\theta_i)\|_2^2 + \mathbb{E}\|\nabla_\theta \hat{J}(\theta_i)\|_2^2$$

We can use Lemma B.1 to bound the first term and Equation 17 to bound the second term. Thus, we have

$$\frac{1}{T}\sum_{i=1}^{T} \mathbb{E}\|\nabla_\theta J(\theta_i)\|_2^2 \leq \frac{2}{T}[TL^2\delta^2 + 2\sqrt{2\Delta_0 L(V+G^2)T}]$$

Substituting the bound for V from Lemma B.3, using the inequality $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ for $a, b \in \mathbb{R}^+$, optimizing over δ, and using $\Delta_0 \leq \mathcal{Q}$ we get

$$\frac{1}{T}\sum_{i=1}^{T} \mathbb{E}\|\nabla_\theta J(\theta_i)\|_2^2 \leq \mathcal{O}(\mathcal{Q}^{\frac{1}{2}} d T^{\frac{-1}{2}} + \mathcal{Q}^{\frac{1}{3}} d^{\frac{2}{3}} T^{\frac{-1}{3}} \sigma)$$

□

C Proof of Theorem

The bound on the bias of the gradient estimate is given by the following lemma:

Lemma C.1. *If the assumptions in Section 4.2 are satisfied, then for the gradient estimate g_i used in Algorithm 4 and the gradient of the objective $J(\theta)$ given in equation 6, we have*

$$\|\mathbb{E}[g_i] - \nabla_\theta J(\theta_i)\| \leq KUH\delta \tag{18}$$

Proof of Lemma C.1. To prove that the bias is bounded, let's consider for any i

$$\begin{aligned}
&\|\mathbb{E}[g_i] - \nabla_\theta J(\theta_i)\|_2 \\
&= \|\sum_{t=0}^{H-1} \mathbb{E}_{s_t \sim d^t_{\pi_{\theta_i}}}[\nabla_\theta \pi(\theta_i, s_t) \\
&\nabla_a(\mathbb{E}_{v\sim\mathbb{B}_p} Q^t_{\pi_{\theta_i}}(s_t, \pi(\theta_i, s_t) + \delta v) - Q^t_{\pi_{\theta_i}}(s_t, \pi(\theta_i, s_t)))]\|_2 \\
&\leq \sum_{t=0}^{H-1} \mathbb{E}_{s_t \sim d^t_{\pi_{\theta_i}}, v\sim\mathbb{B}_p} \|\nabla_\theta \pi(\theta_i, s_t)\|_2 \\
&\quad \|[\nabla_a Q^t_{\pi_{\theta_i}}(s_t, \pi(\theta_i, s_t) + \delta v) - \nabla_a Q^t_{\pi_{\theta_i}}(s_t, \pi(\theta_i, s_t))]\|_2 \\
&\leq \sum_{t=0}^{H-1} KU\delta \mathbb{E}_{v\sim\mathbb{B}_p}\|v\|_2 \\
&\leq KUH\delta
\end{aligned}$$

The first inequality above is obtained by using the fact that $\|\mathbb{E}[X]\|_2 \leq \mathbb{E}\|X\|_2$, and the second inequality using the K-lipschitz assumption on $\pi(\theta, s)$ and U-smooth assumption on $Q^t_{\pi_\theta}(s, a)$ in a. Also, observe that we interchanged the derivative and expectation above by using the assumptions on $Q^t_{\pi_\theta}$ as stated in Section 4.2. □

We will now show that the gradient estimate g_i used in Algorithm 4 has a bounded variance. Note that the gradient estimate constructed in Algorithm 4 is a one-point estimate, unlike policy search in parameter space where we had a two-point estimate. Thus, the variance would be higher and the bound on the variance of such a one-point estimate is given below

Lemma C.2. *Given a gradient estimate g_i as shown in Algorithm 4, the variance of the estimate can be bounded as*

$$\mathbb{E}\|g_i - \mathbb{E}[g_i]\|_2^2 \leq \frac{2H^2p^2K^2}{\delta^2}((\mathcal{Q} + W\delta)^2 + \sigma^2) \tag{19}$$

where σ^2 is the variance of the random noise $\tilde{\eta}$.

Proof of Lemma C.2. To bound the variance of the gradient estimate g_i in Algorithm 4, lets consider

$$\begin{aligned}
&\mathbb{E}_i\|g_i - \mathbb{E}[g_i]\|_2^2 = \mathbb{E}_i\|g_i\|_2^2 - \|\mathbb{E}_i[g_i]\|_2^2 \leq \mathbb{E}_i\|g_i\|_2^2 \\
&= \frac{H^2p^2}{\delta^2}\mathbb{E}_i\|\nabla_\theta\pi(\theta_i, s_t)(Q^t_{\pi_{\theta_i}}(s_t, \pi(\theta_i, s_t) + \delta u) + \tilde{\eta}_i)u\|_2^2 \\
&\leq \frac{K^2p^2H^2}{\delta^2}\mathbb{E}_i\|Q^t_{\pi_{\theta_i}}(s_t, \pi(\theta_i, s_t) + \delta u)u + \tilde{\eta}_i u\|_2^2
\end{aligned}$$

where $\mathbb{E}_i$ denotes expectation with respect to the randomness at iteration i and the inequality is obtained using K-lipschitz assumption on $\pi(\theta, s)$. Note that we can express $Q^t_{\pi_{\theta_i}}(s_t, \pi(\theta_i, s_t)+\delta u) \leq Q^t_{\pi_{\theta_i}}(s_t, \pi(\theta_i, s_t))+W\delta\|u\|_2 \leq \mathcal{Q}+W\delta$ where we used the W-lipschitz assumption on $Q^t_{\pi_\theta}(s, a)$ in a and that it is bounded everywhere by constant $\mathcal{Q}$. Thus, we have

$$\begin{aligned}
&\mathbb{E}_i\|g_i - \mathbb{E}[g_i]\|_2^2 \\
&\leq \frac{K^2p^2H^2}{\delta^2}\mathbb{E}_i\|(\mathcal{Q} + W\delta)u + \tilde{\eta}_i u\|_2^2 \\
&\leq \frac{2K^2p^2H^2}{\delta^2}(\mathbb{E}_i\|(\mathcal{Q} + W\delta)u\|_2^2 + \mathbb{E}_i\|\tilde{\eta}_i u\|_2^2 \\
&\leq \frac{2K^2p^2H^2}{\delta^2}((\mathcal{Q} + W\delta)^2 + \sigma^2)
\end{aligned}$$

□

We are now ready to prove theorem 4.2

Proof of Theorem 4.2. Fix initial solution θ_0 and denote $\Delta_0 = J(\theta_0) - J(\theta^*)$ where θ^* is the point at which $J(\theta)$ attains global minimum. Denote $\xi_i = g_i - \mathbb{E}_i[g_i]$ and $\beta_i = \mathbb{E}_i[g_i] - \nabla_\theta J(\theta_i)$. From Lemma C.1, we know $\|\beta_i\| \leq KUH\delta$ and from lemma C.2, we know $\mathbb{E}\|\xi_i\|_2^2 = V \leq \frac{2K^2p^2H^2}{\delta^2}((\mathcal{Q} + W\delta)^2 + \sigma^2)$ and $\mathbb{E}_i[\xi_i] = 0$ from definition. From $\theta_{i+1} = \theta_i - \alpha g_i$ we have:

$$\begin{aligned}
J(\theta_{i+1}) &= J(\theta_i - \alpha g_i) \\
&\leq J(\theta_i) - \alpha\nabla_\theta J(\theta_i)^T g_i + \frac{L\alpha^2}{2}\|g_i\|_2^2 \\
&= J(\theta_i) - \alpha\nabla_\theta J(\theta_i)^T g_i + \frac{L\alpha^2}{2}\|\xi_i + \mathbb{E}_i[g_i]\|_2^2 \\
&= J(\theta_i) - \alpha\nabla_\theta J(\theta_i)^T g_i \\
&\quad + \frac{L\alpha^2}{2}(\|\mathbb{E}_i[g_i]\|_2^2 + \|\xi_i\|_2^2 + 2\mathbb{E}_i[g_i]^T\xi_i)
\end{aligned}$$

Taking expectation on both sides with respect to randomness at iteration i, we have

$$\begin{aligned}
&\mathbb{E}_i[J(\theta_{i+1})] = J(\theta_i) - \alpha\nabla_\theta J(\theta_i)^T\mathbb{E}_i[g_i] \\
&\quad + \frac{L\alpha^2}{2}(\|\mathbb{E}_i[g_i]\|_2^2 + \mathbb{E}_i\|\xi_i\|_2^2 + 2\mathbb{E}_i[g_i]^T\mathbb{E}_i[\xi_i]) \\
&\leq J(\theta_i) - \alpha\nabla_\theta J(\theta_i)^T(\beta_i + \nabla_\theta J(\theta_i)) \\
&\quad + \frac{L\alpha^2}{2}(\|\beta_i + \nabla_\theta J(\theta_i)\|_2^2 + V) \\
&= J(\theta_i) - \alpha\|\nabla_\theta J(\theta_i)\|_2^2 + \frac{L\alpha^2}{2}(\|\nabla_\theta J(\theta_i)\|_2^2 + V + \|\beta_i\|_2^2) \\
&\quad + (L\alpha^2 - \alpha)\nabla_\theta J(\theta_i)^T\beta_i \\
&\leq J(\theta_i) - \alpha\|\nabla_\theta J(\theta_i)\|_2^2 + \frac{L\alpha^2}{2}(G^2 + V + K^2H^2U^2\delta^2) \\
&\quad + (L\alpha^2 - \alpha)\nabla_\theta J(\theta_i)^T\beta_i
\end{aligned}$$

$$
\begin{aligned}
&\leq J(\theta_i) - \alpha\|\nabla_\theta J(\theta_i)\|_2^2 + \frac{L\alpha^2}{2}(G^2 + V + K^2H^2U^2\delta^2) \\
&\quad + (L\alpha^2 + \alpha)\|\nabla_\theta J(\theta_i)\|\|\beta_i\| \\
&\leq J(\theta_i) - \alpha\|\nabla_\theta J(\theta_i)\|_2^2 + \frac{L\alpha^2}{2}(G^2 + V + K^2H^2U^2\delta^2) \\
&\quad + (L\alpha^2 + \alpha)GKUH\delta
\end{aligned}
$$

Rearranging terms and summing over timestep 1 to T, we get

$$
\begin{aligned}
\alpha\sum_{i=1}^{T}\|\nabla_\theta J(\theta_i)\|_2^2 &\leq J(\theta_0) - \mathbb{E}_T[J(\theta_T)] \\
&\quad + \frac{LT\alpha^2}{2}(G^2 + V + K^2H^2U^2\delta^2) + (L\alpha^2 + \alpha)GKUHT\delta \\
&\leq \Delta_0 + \frac{LT\alpha^2}{2}(G^2 + V + K^2H^2U^2\delta^2) \\
&\quad + (L\alpha^2 + \alpha)GKUHT\delta \\
\sum_{i=1}^{T}\|\nabla_\theta J(\theta_i)\|_2^2 &\leq \frac{\Delta_0}{\alpha} + \frac{LT\alpha}{2}(G^2 + V + K^2H^2U^2\delta^2) \\
&\quad + (L\alpha + 1)GKUHT\delta \\
&\leq \frac{\Delta_0}{\alpha} + \frac{LT\alpha}{2}(G^2 + K^2H^2U^2\delta^2 + 2GKUH\delta) \\
&\quad + GKUHT\delta + \frac{LT\alpha}{2}V \\
&\leq \frac{\Delta_0}{\alpha} + \frac{LT\alpha}{2}(G + KHU\delta)^2 \\
&\quad + GKUHT\delta + \frac{LT\alpha K^2p^2H^2}{\delta^2}((\mathcal{Q} + W\delta)^2 + \sigma^2) \\
&\leq \frac{\Delta_0}{\alpha} + LT\alpha(G^2 + K^2H^2U^2\delta^2) \\
&\quad + GKUHT\delta + 2\frac{LT\alpha K^2p^2H^2}{\delta^2}(\mathcal{Q}^2 + W^2\delta^2 + \sigma^2)
\end{aligned}
$$

Using $\Delta_0 \leq \mathcal{Q}$ and optimizing over α and δ, we get $\alpha = \mathcal{O}(\mathcal{Q}^{\frac{3}{4}}T^{-\frac{3}{4}}H^{-1}p^{-\frac{1}{2}}(\mathcal{Q}^2+\sigma^2)^{-\frac{1}{4}})$ and $\delta = \mathcal{O}(T^{-\frac{1}{4}}p^{\frac{1}{2}}(\mathcal{Q}^2+\sigma^2)^{\frac{1}{4}})$. This gives us

$$
\frac{1}{T}\sum_{i=1}^{T}\|\nabla_\theta J(\theta_i)\|_2^2 \leq \mathcal{O}(T^{-\frac{1}{4}}Hp^{\frac{1}{2}}(\mathcal{Q}^3 + \sigma^2\mathcal{Q})^{\frac{1}{4}}) \tag{20}
$$

□

D Implementation Details

D.1 One-step Control Experiments

D.1.1 Tuning Hyperparameters for ARS

We tune the hyperparameters for ARS (Mania et al., 2018) in both MNIST and linear regression experiments, by choosing a candidate set of values for each hyperparameter: stepsize, number of directions sampled, number of top directions chosen and the perturbation length along each direction. The candidate hyperparameter values are shown in Table 1.

We use the hyperparameters shown in Table 2 chosen through this tuning for each of the experiments in this work. The hyperparameters are chosen by averaging the test squared loss across three random seeds (different from the 10 random seeds used in actual experiments) and chosing the setting that has the least mean test squared loss after 100000 samples.

Hyperparameter	**Candidate Values**
Stepsize	$0.001, 0.005, 0.01, 0.02, 0.03$
# Directions	$10, 50, 100, 200, 500$
# Top Directions	$5, 10, 50, 100, 200$
Perturbation	$0.001, 0.005, 0.01, 0.02, 0.03$

Table 1: Candidate hyperparameters used for tuning in ARS experiments

Experiment	**Stepsize**	**# Dir.**	**# Top Dir.**	**Perturbation**
MNIST	0.02	50	20	0.03
LR $d = 10$	0.03	10	10	0.03
LR $d = 100$	0.03	10	10	0.02
LR $d = 1000$	0.03	200	200	0.03

Table 2: Hyperparameters chosen for ARS in each experiment. LR is short-hand for Linear Regression.

D.1.2 MNIST Experiments

The CNN architecture used is as shown in Figure 4[1]. The total number of parameters in this model is $d = 21840$. For supervised learning, we use a cross-entropy loss on the softmax output with respect to the true label. To train this model, we use a batch size of 64 and a stochastic gradient descent (SGD) optimizer with learning rate of 0.01 and a momentum factor of 0.5. We evaluate the test accuracy of the model over all the 10000 images in the MNIST test dataset.

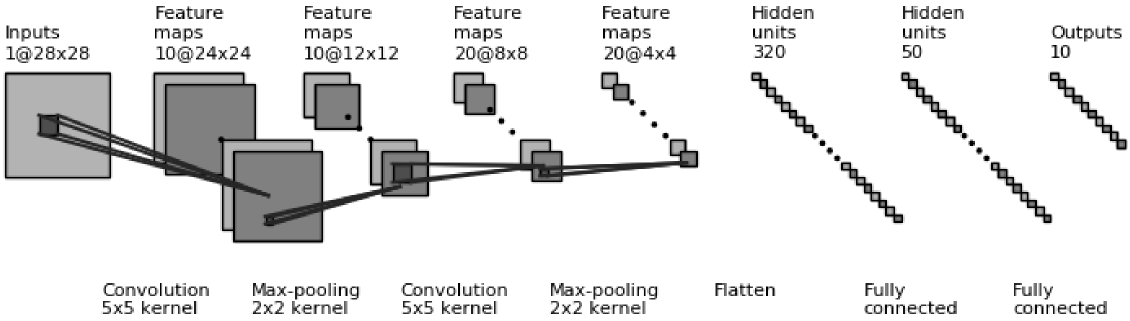

Figure 4: CNN architecture used for the MNIST experiments

For REINFORCE, we use the same architecture as before. We train the model by sampling from the categorical distribution parameterized by the softmax output of the model and then computing a ±1 reward based on whether the model predicted the correct label. The loss function is the REINFORCE loss function given by,

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N} r_i \log(\mathbb{P}(\hat{y}_i | x_i, \theta)) \tag{21}$$

where θ is the parameters of the model, r_i is the reward obtained for example i, $\hat{y}_i$ is the predicted label for example i and x_i is the input feature vector for example i. The reward r_i is given by $r_i = 2 * \mathbb{I}[\hat{y}_i = y_i] - 1$, where $\mathbb{I}$ is the $0-1$ indicator function and y_i is the true label for example i.

For ARS, we use the same architecture and reward function as before. The hyperparameters used are shown in Table 2 and we closely follow the algorithm outlined in (Mania et al., 2018).

[1] This figure is generated by adapting the code from `https://github.com/gwding/draw_convnet`

Experiment	Learning Rate	Batch size
MNIST	0.001	512
LR $d = 10$	0.08	512
LR $d = 100$	0.03	512
LR $d = 1000$	0.01	512

Table 3: Learning rate and batch size used for REINFORCE experiments. We use an ADAM (Kingma and Ba, 2014) optimizer for these experiments.

Experiment	Learning Rate	Batch size
LR $d = 10$	2.0	512
LR $d = 100$	2.0	512

Table 4: Learning rate and batch size used for Natural REINFORCE experiments. Note that we decay the learning rate after each batch by $\sqrt{T}$ where T is the number of batches seen.

D.1.3 Linear Regression Experiments

We generate training and test data for the linear regression experiments as follows: we sampled a random $d+1$ dimensional vector w where d is the input dimensionality. We also sampled a random $d \times d$ covariance matrix C. The training and test dataset consists of $d+1$ vectors x whose first element is always 1 (for the bias term) and the rest of the d terms are sampled from a multivariate normal distribution with mean $\mathbf{0}$ and covariance matrix C. The target vectors y are computed as $y = w^T x + \epsilon$ where ϵ is sampled from a univariate normal distribution with mean 0 and standard deviation 0.001.

We implemented both SGD and Newton Descent on the mean squared loss, for the supervised learning experiments. For SGD, we used a learning rate of 0.1 for $d = 10, 100$ and a learning rate of 0.01 for $d = 1000$, and a batch size of 64. For Newton Descent, we also used a batch size of 64. To frame it as a one-step MDP, we define a reward function r which is equal to the negative of mean squared loss. Both REINFORCE and ARS use this reward function. To compute the REINFORCE loss, we take the prediction of the model $\hat{w}^T x$, add a mean 0 standard deviation $\beta = 0.5$ Gaussian noise to it, and compute the reward (negative mean squared loss) for the noise added prediction. The REINFORCE loss function is then given by

$$J(w) = \frac{1}{N} \sum_{i=1}^{N} r_i \frac{-(y_i - \hat{w}^T x_i)^2}{2\beta^2} \tag{22}$$

where $r_i = -(y_i - \hat{y}_i)^2$, $\hat{y}_i$ is the noise added prediction and $\hat{w}^T x_i$ is the prediction by the model. We use an Adam optimizer with learning rate and batch size as shown in Table 3. For the natural REINFORCE experiments, we estimate the fisher information matrix and compute the descent direction by solving the linear system of equations $Fx = g$ where F is the fisher information matrix and g is the REINFORCE gradient. We use SGD with a $O(1/\sqrt{T})$ learning rate, where T is the number of batches seen, and batch size as shown in Table 4.

For ARS, we closely follow the algorithm outlined in (Mania et al., 2018).

D.2 Multi-step Control Experiments

D.2.1 Tuning Hyperparameters for ARS

We tune the hyperparameters for ARS (Mania et al., 2018) in both mujoco and LQR experiments, similar to the one-step control experiments. The candidate hyperparameter values are shown in Tables 5 and 6. We have observed that using all the directions in ARS is always preferable under the low horizon settings that we explore. Hence, we do not conduct a hyperparameter search over the number of top directions and instead keep it the same as the number of directions.

We use the hyperparameters shown in Tables 7 and 8 chosen through tuning for each of the multi-step experiments. The hyperparameters are chosen by averaging the total reward obtained across three random seeds (different from the 10 random seeds used in experiments presented in Figures 3a, 3b, 3c) and chosing the setting that has the highest total reward after 10000 episodes of training..

Hyperparameter	**Swimmer-v2**	**HalfCheetah-v2**
Stepsize	$0.03, 0.05, 0.08, 0.1, 0.15$	$0.001, 0.003, 0.005, 0.008, 0.01$
# Directions	$5, 10, 20$	$5, 10, 20$
Perturbation	$0.05, 0.1, 0.15, 0.2$	$0.01, 0.03, 0.05, 0.08$

Table 5: Candidate hyperparameters used for tuning in ARS experiments

Hyperparameter	**LQR**
Stepsize	$0.0001, 0.0003, 0.0005, 0.0008, 0.001, 0.003, 0.005, 0.008, 0.01$
# Directions	10
Perturbation	$0.01, 0.05, 0.1$

Table 6: Candidate hyperparameters used for tuning in ARS experiments

D.2.2 Tuning Hyperparameters for ExAct

We tune the hyperparameters for ExAct (Algorithm 4) in both mujoco and LQR experiments, similar to ARS. The candidate hyperparameter values are shown in Tables 9 and 10. Similar to ARS, we do not conduct a hyperparameter search over the number of top directions and instead keep it the same as the number of directions.

Hyperparameter	**Swimmer-v2**	**HalfCheetah-v2**
Stepsize	$0.005, 0.008, 0.01, 0.015, 0.02, 0.025, 0.03$	$0.0001, 0.0003, 0.0005, 0.0008, 0.001, 0.002, 0.003$
# Directions	$5, 10, 20$	$5, 10, 20$
Perturbation	$0.15, 0.2, 0.3, 0.5$	$0.15, 0.2, 0.3, 0.5$

Table 9: Candidate hyperparameters used for tuning in ExAct experiments

We use the hyperparameters shown in Tables 11 and 12 chosen through tuning for each of the multi-step experiments, similar to ARS.

D.2.3 Mujoco Experiments

For all the mujoco experiments, both ARS and ExAct use a linear policy with the same number of parameters as the dimensionality of the state space. The hyperparameters for both algorithms are chosen as described above. Each algorithm is run on both environments (Swimmer-v2 and HalfCheetah-v2) for 10000 episodes of training across 10 random seeds (different from the ones used for tuning). This is repeated for each horizon value $H \in \{1, 2, \cdots, 15\}$. In each experiment, we record the mean evaluation return obtained after training and plot the results in Figures 3a, 3b. For more details on the environments used, we refer the reader to (Brockman et al., 2016b).

D.2.4 LQR Experiments

In the LQR experiments, we constructed a linear dynamical system $x_{t+1} = Ax_t + Bu_t + \xi_t$ where $x_t \in \mathbb{R}^{100}$, $A \in \mathbb{R}^{100\times100}$, $B \in \mathbb{R}^{100}$, $u_t \in \mathbb{R}$ and the noise $\xi_t \sim \mathcal{N}(0_{100}, cI_{100\times100})$ with a small constant $c \in \mathbb{R}^+$. We explicitly make sure that the maximum eigenvalue of A is less than 1 to avoid instability. We fix a quadratic cost function $c(x, u) = x^TQx + uRu$, where $Q = 10^{-3}I_{100\times100}$ and $R = 1$. The hyperparameters chosen for both algorithms are chosen as described above.

For each algorithm, we run it for noise covariance values $c \in \{10^{-4}, 5\times10^{-4}, 10^{-3}, 5\times10^{-3}, 10^{-2}, 5\times10^{-2}, 10^{-1}, 5\times10^{-1}\}$ until we reach a stationary point where $\|\nabla_\theta J(\theta)\|_2^2 \leq 0.05$. The number of interactions with the environment allowed is capped at 10^6 steps for each run. This is repeated across 10 random seeds (different from the ones used for tuning). The number of interactions needed to reach the stationary point as the noise covariance is increased is recorded and shown in Figure 3c.

Horizon	**Stepsize**	**# Directions**	**Perturbation**
$H = 1$	0.15	5	0.2
$H = 2$	0.08	5	0.2
$H = 3$	0.15	5	0.2
$H = 4$	0.08	5	0.2
$H = 5$	0.05	5	0.2
$H = 6$	0.08	5	0.2
$H = 7$	0.08	5	0.2
$H = 8$	0.08	5	0.2
$H = 9$	0.1	5	0.2
$H = 10$	0.08	5	0.2
$H = 11$	0.08	5	0.2
$H = 12$	0.1	5	0.2
$H = 13$	0.08	5	0.2
$H = 14$	0.08	5	0.2
$H = 15$	0.08	10	0.2

Table 7: Hyperparameters chosen for multi-step experiments for ARS in Swimmer-v2

Horizon	**Stepsize**	**# Directions**	**Perturbation**
$H = 1$	0.001	20	0.08
$H = 2$	0.008	5	0.08
$H = 3$	0.008	10	0.08
$H = 4$	0.003	5	0.05
$H = 5$	0.003	5	0.05
$H = 6$	0.003	10	0.05
$H = 7$	0.008	20	0.05
$H = 8$	0.008	5	0.05
$H = 9$	0.01	20	0.03
$H = 10$	0.005	10	0.03
$H = 11$	0.008	20	0.03
$H = 12$	0.005	5	0.05
$H = 13$	0.008	20	0.03
$H = 14$	0.01	10	0.03
$H = 15$	0.008	20	0.03

Table 8: Hyperparameters chosen for multi-step experiments for ARS in HalfCheetah-v2

Hyperparameter	**LQR**
Stepsize	$0.0001, 0.0003, 0.0005, 0.0008, 0.001, 0.003, 0.005, 0.008, 0.01$
# Directions	10
Perturbation	$0.01, 0.05, 0.1$

Table 10: Candidate hyperparameters used for tuning in ExAct experiments

Horizon	**Stepsize**	**# Directions**	**Perturbation**
$H = 1$	0.02	5	0.2
$H = 2$	0.02	5	0.2
$H = 3$	0.015	10	0.2
$H = 4$	0.015	10	0.2
$H = 5$	0.01	10	0.2
$H = 6$	0.015	10	0.2
$H = 7$	0.01	20	0.2
$H = 8$	0.015	20	0.2
$H = 9$	0.02	20	0.2
$H = 10$	0.008	5	0.2
$H = 11$	0.02	5	0.15
$H = 12$	0.02	20	0.2
$H = 13$	0.015	5	0.15
$H = 14$	0.02	10	0.15
$H = 15$	0.01	5	0.1

Table 11: Hyperparameters chosen for multi-step experiments for ExAct in Swimmer-v2

Horizon	**Stepsize**	**# Directions**	**Perturbation**
$H = 1$	0.0001	20	0.2
$H = 2$	0.001	5	0.2
$H = 3$	0.001	5	0.2
$H = 4$	0.001	5	0.2
$H = 5$	0.001	10	0.2
$H = 6$	0.001	5	0.2
$H = 7$	0.001	10	0.2
$H = 8$	0.001	5	0.2
$H = 9$	0.001	5	0.2
$H = 10$	0.001	5	0.2
$H = 11$	0.0008	5	0.15
$H = 12$	0.001	5	0.2
$H = 13$	0.001	10	0.2
$H = 14$	0.001	5	0.2
$H = 15$	0.0008	10	0.2

Table 12: Hyperparameters chosen for multi-step experiments for ExAct in HalfCheetah-v2