Variance Reduction for Evolutionary Strategies via Structured Control Variates

Yunhao Tang
Columbia University

Krzysztof Choromanski
Google Robotics

Alp Kucukelbir
Fero Labs & Columbia University

Abstract

Evolution strategies (ES) are a powerful class of blackbox optimization techniques that recently became a competitive alternative to state-of-the-art policy gradient (PG) algorithms for reinforcement learning (RL). We propose a new method for improving accuracy of the ES algorithms, that as opposed to recent approaches utilizing only Monte Carlo structure of the gradient estimator, takes advantage of the underlying Markov decision process (MDP) structure to reduce the variance. We observe that the gradient estimator of the ES objective can be alternatively computed using reparametrization and PG estimators, which leads to new control variate techniques for gradient estimation in ES optimization. We provide theoretical insights and show through extensive experiments that this RL-specific variance reduction approach outperforms general purpose variance reduction methods.

1 Introduction

Evolution strategies (ES) have regained popularity through their successful application to modern reinforcement learning (RL). ES are a powerful alternative to policy gradient (PG) methods. Instead of leveraging the Markov decision process (MDP) structure of a given RL problem, ES cast the RL problem as a blackbox optimization. To carry out this optimization, ES use gradient estimators based on randomized finite difference methods. This presents a trade-off: ES are better at handling long term horizons and sparse rewards than PG methods, but the ES gradient estimator may exhibit prohibitively large variance.

Variance reduction techniques can make both methods more practical. Control variates (also known as baseline functions) that leverage Markovian (Mnih et al., 2016; Schulman et al., 2015b, 2017) and factorized policy structures (Gu et al., 2016; Liu et al., 2017; Grathwohl et al., 2018; Wu et al., 2018), help to improve PG methods. In contrast to these structured approaches, variance reduction for ES has been focused on general-purpose Monte Carlo (MC) techniques, such as antithetic sampling (Salimans et al., 2017; Mania et al., 2018), orthogonalization (Choromanski et al., 2018), optimal couplings (Rowland et al., 2018) and quasi-MC sampling (Choromanski et al., 2018; Rowland et al., 2018).

Main idea. We propose a variance reduction technique for ES that leverages the underlying MDP structure of RL problems. We begin with a simple reparameterization of the problem that uses PG estimators computed via backpropagation. We follow by constructing a control variate using the difference of two gradient estimators of the same objective. The result is a RL-specific variance reduction technique for ES that achieves better performance across a wide variety of RL problems.

Figure 1 summarizes the performance of our proposal over 16 RL benchmark tasks. Our method consistently improves over vanilla ES baselines and other state-of-the-art, general purpose MC variance reduction methods. Moreover, we provide theoretical insight into why our algorithm achieves more variance reduction than orthogonal ES (Choromanski et al., 2018) when the policy itself is highly stochastic. (Section 3 presents a detailed analysis.)

Related Work. Control variates are commonly used to reduce the variance of MC-based estimators (Ross, 2002). In black-box variational inference algorithms, carefully designed control variates can reduce the variance of MC gradient updates, leading to faster convergence (Paisley et al., 2012; Ranganath et al., 2014). In RL, PG methods apply state-dependent baseline functions as control variates (Mnih et al., 2016; Schulman et al., 2015b, 2017). While action-dependent control
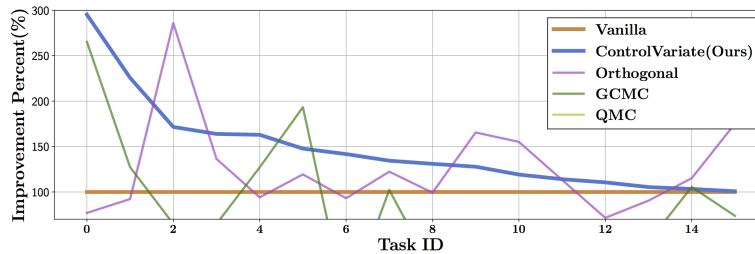
Figure 1: Percentile improvement over 16 benchmark tasks. The metric is calculated for each task as $(r_{\text{cv}} - r_{\text{random}})/(r_{\text{es}} - r_{\text{random}})$, where $r_{\text{cv}}, r_{\text{es}}, r_{\text{random}}$ are the final rewards for the RL tasks obtained with our control variate, vanilla ES, and random policy methods respectively. We see that our proposal consistently improves over vanilla ES for all tasks, and over all compared variance reduction methods for 10 out of 16 tasks. Section 4 provides additional details.

variates (Gu et al., 2017; Liu et al., 2017; Grathwohl et al., 2018; Wu et al., 2018) have been proposed to achieve further variance reduction, Tucker et al. (2018) recently showed that the reported performance gains may be due to subtle implementation details, rather than better baseline functions.

To leverage the MDP structure in developing a RL-specific control variate for ES, we derive a gradient estimator for the ES objective based on reparameterization (Kingma and Welling, 2014) and the PG estimator. The control variate is constructed as a difference between two alternative gradient estimators. Our approach is related to a control variate techniques developed for modeling discrete latent variables in variational inference (Tucker et al., 2017). The idea is to relax the discrete model into a differentiable one and construct the control variate as the difference between the score function gradient estimator and reparameterized gradient estimator of the relaxed model. We expand on this connection in Section 3.3.

2 Policy Optimization in Reinforcement Learning

Sequential decision making problems are often formulated as a MDPs. Consider an episode indexed by time. At any given time $t \geq 0$, an agent is in a state $s_t \in \mathcal{S}$. The agent then takes an action $a_t \in \mathcal{A}$, receives an instant reward $r_t = r(s_t, a_t) \in \mathbb{R}$, and transitions to the next state $s_{t+1} \sim p(\cdot \mid s_t, a_t)$, where p is a distribution determining transitional probabilities. Define the policy $\pi : \mathcal{S} \mapsto \mathcal{P}(\mathcal{A})$ as a conditional distribution over actions $\mathcal{A}$ given a state $s \in \mathcal{S}$. RL seeks to maximize the expected cumulative rewards over a given time horizon T,

$$J^\gamma(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{T-1} r_t \gamma^t \right], \tag{1}$$

where $\gamma \in (0, 1]$ is a discount factor and the expectation is with respect to randomized environment and outputs of the policy π.

Ideally, we would like to work with an infinite horizon and no discount factor. In practice, horizon T is bounded by sample collection (Brockman et al., 2016) while directly optimizing the undiscounted objective $J^1(\pi)$ admits unusably high variance gradient estimators (Schulman et al., 2015b). As a result, modern RL algorithms tackle the problem through a discount factor $\gamma < 1$, which reduces the variance of the gradient estimators but introduces bias (Schulman et al., 2015b, 2017; Mnih et al., 2016). At evaluation time, the algorithms are evaluated with finite horizons $T < \infty$ and undiscounted returns $\gamma = 1$ (Schulman et al., 2015b, 2017; Mnih et al., 2016). We follow this setup here as well.

Consider parameterizing the policy as π_θ where $\theta \in \mathbb{R}^d$. The goal is to optimize Equation (1) with respect to policy parameters. A natural approach is to use exact gradient methods. Regrettably, this objective function does not admit an analytic gradient. Thus, we turn to stochastic gradient techniques (Robbins and Monro, 1951) and seek to construct approximations to the true gradient $g_\theta^\gamma = \nabla_\theta J^\gamma(\pi_\theta)$.

2.1 Evolution Strategies for Policy Optimization

Evolution strategies (ES) (Salimans et al., 2017) take a black-box optimization approach to maximizing Equation (1). To do so, the first step is to ensure that the objective function is differentiable with respect to the policy parameters. To this end, ES begin by convolving the original objective $J^\gamma(\pi_\theta)$ with a multivariate isotropic Gaussian distribution of mean θ and variance σ^2:

$$F^{\sigma,\gamma}(\theta) = \mathbb{E}_{\theta' \sim \mathcal{N}(\theta, \sigma^2 \mathbb{I})} \left[J^\gamma(\pi_{\theta'}) \right]. \tag{2}$$

ES maximize this smooth objective as a proxy to maximizing the original objective $J^\gamma(\pi_\theta)$. The convolved objective F enjoys the advantage of being differentiable with respect to the policy. In the limit $\sigma \to 0$, an optimal point of $F^{\sigma,\gamma}(\theta)$ is also optimal with respect to $J^\gamma(\pi_\theta)$. The next step is to derive a gradient of Equation (2). Consider the score function gradient,

$$\nabla_\theta F^{\sigma,\gamma}(\theta) = \mathbb{E}_{\theta' \sim \mathcal{N}(\theta,\sigma^2\mathbb{I})}\left[J^\gamma(\pi_{\theta'})\frac{\theta' - \theta}{\sigma^2}\right]. \quad (3)$$

This gradient can be computed by sampling $\theta'_i \sim \mathcal{N}(\theta, \sigma^2\mathbb{I})$ and computing unbiased estimates of each $J^\gamma(\pi_{\theta'_i})$ using a single roll-out trajectory of $\pi_{\theta'_i}$ in the environment. The resulting score function gradient estimator has the following form:

$$\hat{g}_\theta^{\text{ES},\gamma} = \frac{1}{N}\sum_{i=1}^N J^\gamma(\pi_{\theta'_i})\frac{\theta'_i - \theta}{\sigma^2}. \quad (4)$$

This gradient estimator is biased with respect to the original objective. However, in practice this bias does not hinder optimization; on the contrary, the smoothed objective landscape is often more amenable to gradient-based optimization (Leordeanu and Hebert, 2008). We also make clear that though the ES gradient is defined for any $\gamma \in (0,1]$, in practice parameters are updated with the gradient of the undiscounted objective $\hat{g}_\theta^{\text{ES},1}$ (Salimans et al., 2017; Mania et al., 2018; Choromanski et al., 2018).

2.2 Policy Gradient Methods for Policy Optimization

Policy gradient (PG) (Sutton et al., 2000) methods take a different approach. Instead of deriving the gradient through a parameter level perturbation as in ES, the core idea of PG is to leverage the randomness in the policy itself. Using a standard procedure from stochastic computational graphs (Schulman et al., 2015a), we compute the gradient of Equation (1) as follows

$$\nabla_\theta J^\gamma(\pi_\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{T-1}\left(\sum_{t'=t}^{T-1} r_{t'}\right)\gamma^t \nabla_\theta \log \pi_\theta(a_t|s_t)\right]. \quad (5)$$

Unbiased estimators $\hat{g}_\theta^{\text{PG},\gamma}$ of this gradient can be computed using sampling as above for the ES method. In practice, the sample estimate of Equation (5) often has large variance which destabilizes the updates. To alleviate this issue, one convenient choice is to set $\gamma < 1$ so that the long term effects of actions are weighted down exponentially. This reduces the variance of the estimator, but introduces bias with respect to the original undiscounted objective $J^1(\pi_\theta)$.

ES and PG are two alternative methods for deriving gradient estimators with respect to the policy parameters. On an intuitive level, these two methods complement each other for variance reduction: PG leverages the MDP structure and achieves lower variance when the policy is stochastic; ES derives the gradient by injecting noise directly into the parameter space and is characterized by lower variance when the policy itself is near-deterministic. Our goal in this paper is to develop a single estimator that benefits from both approaches. We formalize this intuition in the next section.

3 Variance Reduction via Structured Control Variates

We seek a control variate for the ES gradient estimator in Equation (4). Recall that this gradient is with respect to a smoothed objective: $F^{\sigma,\gamma}(\theta)$ with $\gamma \in (0,1]$.

3.1 Reparameterized Gradients of the Smoothed Objective

The ES gradient estimator in Equation (4) leverages the derivative of the logarithm. We can also apply the reparameterization technique (Kingma and Welling, 2014) to the distribution $\theta' \sim \mathcal{N}(\theta, \sigma^2\mathbb{I})$ to obtain:

$$\begin{aligned}\nabla_\theta F^{\sigma,\gamma}(\theta) &= \nabla_\theta \mathbb{E}_{\theta' \sim \mathcal{N}(\theta,\sigma^2\mathbb{I})}[J^\gamma(\pi_{\theta'})] \\ &= \nabla_\theta \mathbb{E}_{\epsilon \sim \mathcal{N}(0,\mathbb{I})}[J^\gamma(\pi_{\theta+\epsilon\cdot\sigma})] \\ &= \mathbb{E}_{\epsilon \sim \mathcal{N}(0,\mathbb{I})}[\nabla_{\theta+\epsilon\cdot\sigma}J^\gamma(\pi_{\theta+\epsilon\cdot\sigma})], \quad (6)\end{aligned}$$

where $\nabla_{\theta+\epsilon\cdot\sigma}J^\gamma(\pi_{\theta+\epsilon\cdot\sigma})$ can be computed by PG estimators for the discounted objective (5). To estimate (6), we sample $\epsilon_i \sim \mathcal{N}(0, \sigma^2\mathbb{I})$ and construct perturbed policies $\theta'_i = \theta + \epsilon_i \cdot \sigma$. Then an unbiased estimate $\hat{g}_{\theta+\epsilon_i\cdot\sigma}^{\text{PG},\gamma}$ of the policy gradient $\nabla_{\theta+\epsilon_i\cdot\sigma}J^\gamma(\pi_{\theta+\epsilon_i\cdot\sigma})$ can be computed from a single rollout trajectory using $\pi_{\theta+\epsilon_i\cdot\sigma}$. Finally the reparameterized gradient is computed by averaging:

$$\hat{g}_\theta^{\text{RE},\gamma} = \frac{1}{N}\sum_{i=1}^N \hat{g}_{\theta+\epsilon_i\cdot\sigma}^{\text{PG},\gamma}. \quad (7)$$

3.2 Evolution Strategies with Structured Control Variates

For the discounted objective $F^{\sigma,\gamma}(\theta)$ we have two alternative gradient estimators. One is constructed using the score function gradient estimator (see: Equation (4)). The other uses the re-parameterization techqniue along with policy gradient estimators (see: Equation (7)). Combining these two estimators with the vanilla ES gradient for the undiscounted objective

$\hat{g}_\theta^{\text{ES},1}$, we get:

$$\hat{g}_\theta^{\text{CV}} = \hat{g}_\theta^{\text{ES},1} + \eta \odot (\hat{g}_\theta^{\text{ES},\gamma} - \hat{g}_\theta^{\text{RE},\gamma}), \qquad (8)$$

where η is a vector of same dimension of θ and $\odot$ denotes an element-wise product. This scaling parameter η controls the relative importance of the two terms in (8). As discussed below, we can adapt the discount factor γ and the scaling parameter η to minimize the variance over time.

Discount factor γ. As in Choromanski et al. (2018), for a vector $g \in \mathbb{R}^n$, we define its variance as the sum of its component variances $\mathbb{V}[g] \coloneqq \sum_{i=1}^n \mathbb{V}[g_i]$. We then adapt the discount factor $\gamma \leftarrow \gamma - \alpha_\gamma \nabla_\gamma \mathbb{V}[\hat{g}_\theta^{\text{CV}}]$ for some learning rate $\alpha_\gamma > 0$. Since $\mathbb{E}[\hat{g}_\theta^{\text{CV}}]$ does not depend on γ, we have equivalently $\nabla_\gamma \mathbb{V}[\hat{g}_\theta^{\text{CV}}] = \nabla_\gamma \mathbb{E}[(\hat{g}_\theta^{\text{CV}})^2]$. The gradient $\nabla_\gamma \mathbb{E}[(\hat{g}_\theta^{\text{CV}})^2]$ can be itself estimated using backpropagation on mini-batches but this tends to be unstable because each term in (5) involves γ^t. Alternatively, we build a more robust estimator of $\nabla_\gamma \mathbb{E}[(\hat{g}_\theta^{\text{CV}})^2]$ using ES: in particular, sample $\epsilon_i \sim \mathcal{N}(0,1)$ and let v_i be the evaluation of $\mathbb{E}[(\hat{g}_\theta^{\text{CV}})^2]$ under $\gamma + \sigma_\gamma \epsilon_i$ for some $\sigma_\gamma > 0$. The gradient estimator for γ is $\hat{g}_\gamma = \frac{1}{N} \sum_{i=1}^N v_i \frac{\epsilon_i}{\sigma_\gamma}$.

Though the full estimator (8) is defined for all discount factors $\gamma \in (0,1]$, in general we find it better to set $\gamma < 1$ to stablize the PG components of the control variate.

Coefficient η. Since η is a vector with the same dimensionality as θ, we can update each component of η to reduce the variance of each component of $\hat{g}_\theta^{\text{CV}}$. Begin by computing, $\nabla_\eta \mathbb{V}[\hat{g}_\theta^{\text{CV}}]$ as follows:

$$\nabla_\eta \mathbb{V}\left[\hat{g}_\theta^{\text{CV}}\right] = 2\eta \odot \mathbb{E}\left[\left(\hat{g}_\theta^{\text{ES},\gamma} - \hat{g}_\theta^{\text{RE},\gamma}\right)^2\right] +$$
$$2\mathbb{E}\left[\left(\hat{g}_\theta^{\text{ES},\gamma} - \hat{g}_\theta^{\text{RE},\gamma}\right) \odot \hat{g}_\theta^{\text{ES},1}\right]. \qquad (9)$$

Then, estimate this gradient using MC sampling. Finally, adapt η by running online gradient descent: $\eta \leftarrow \eta - \alpha_\eta \nabla_\eta \mathbb{V}[\hat{g}_\theta^{\text{CV}}]$ with some $\alpha_\eta > 0$.

Practical considerations. Certain practical techniques can be applied to stabilize the ES optimization procedure. For example, Salimans et al. (2017) apply a centered rank transformation to the estimated returns $J^\gamma(\pi_{\theta_i'})$ to compute the estimator of the gradient in Equation (4). This transformation is compatible with our proposal. The construction becomes $\hat{g}_\theta^{\text{CV}} = \hat{g}_\theta + \eta(\hat{g}_\theta^{\text{ES},\gamma} - \hat{g}_\theta^{\text{RE},\gamma})$ where $\hat{g}_\theta$ can be computed through the rank transformation.

Stochastic policies. While many prior works (Salimans et al., 2017; Mania et al., 2018; Choromanski

et al., 2018; Rowland et al., 2018) focus on deterministic policies for continuous action spaces, our method targets stochastic policies, as required by the PG computation. Estimating PG for a deterministic policy requires training critic functions and is in general biased (Silver et al., 2014). We leave the investigation of deterministic policies to future work.

3.3 Relationship to REBAR

REBAR (Tucker et al., 2017) considers variance reduction of gradient estimators for probabilistic models with discrete latent variables. The discrete latent variable model has a *relaxed model* version, where the discrete sampling procedure is replaced by a differentiable function with reparameterized noise. This relaxed model usually has a temperature parameter τ such that when $\tau \to 0$ the relaxed model converges to the original discrete model. To optimize the discrete model, the baseline approach is to use score function gradient estimator, which is unbiased but has high variance. Alternatively, one could use the reparameterized gradient through the relaxed model, which has lower variance, but the gradient is biased for finite $\tau > 0$ (Jang et al., 2017; Maddison et al., 2017). The bias and variance of the gradient through the relaxed model is controlled by τ. REBAR proposes to use the difference between the score function gradient and reparameterized gradient of the relaxed model as a control variate. Their difference has expectation zero and should be highly correlated with the reinforced gradient of the original discrete model, leading to potentially large variance reduction.

A similar connection can be found in ES for RL context. We can interpret the non-discounted objective, namely $F^{\sigma,\gamma}(\pi_\theta)$ with $\gamma = 1$, as the original model which gradient we seek to estimate. When $\gamma < 1$, we have the relaxed model which gradient becomes biased but has lower variance (with respect to the non-discounted objective). Similar to REBAR, our proposal is to construct the score function gradient (ES estimator) $\hat{g}_\theta^{\text{ES},\gamma}$ and reparameterized gradient (PG estimator) $\hat{g}_\theta^{\text{RE},\gamma}$ of the general discounted objective $\gamma < 1$ (relaxed model), such that their difference $\hat{g}_\theta^{\text{ES},\gamma} - \hat{g}_\theta^{\text{RE},\gamma}$ serves as a control variate. The variance reduction from REBAR applies here, $\hat{g}_\theta^{\text{ES},\gamma} - \hat{g}_\theta^{\text{RE},\gamma}$ should be highly correlated with $\hat{g}_\theta^{\text{ES},1}$, which leads to effective variance reduction.

3.4 How much variance reduction is possible?

How does the variance reduction provided by control variate compare to that of general purpose methods, such as orthogonalization (Choromanski et al., 2018)? In this section, we build on a simple example to illustrate the different variance reduction properties of these approaches. Recall that we define the variance

Yunhao Tang, Krzysztof Choromanski, Alp Kucukelbir

of a vector $g \in \mathbb{R}^d$ as the sum of the variance of its components $\mathbb{V}[g] = \sum_{i=1}^{d} \mathbb{V}[g_i]$ following notation from prior literature (Choromanski et al., 2018; Rowland et al., 2018).

Consider a one-step MDP problem where the agent takes only one action a and receives a reward $r(a) = \alpha^T a$ for some $\alpha \in \mathbb{R}^d$. We choose the reward function to be linear in a, as a local approximation to a potentially nonlinear reward function landscape. Let the policy be a Gaussian with mean μ and diagonal covariance matrix $\Sigma = \sigma_2^2 \mathbb{I}$ with fixed σ_2. Here the policy parameter contains only the mean $\theta = \mu$. The RL objective is $J(\pi_\mu) = \mathbb{E}_{a \sim \mathcal{N}(\mu, \Sigma)}[r(a)]$. To compute the gradient, ES smoothes the objective with the Gaussin $\mathcal{N}(\mu, \sigma_1^2 \mathbb{I})$ for a fixed σ_1. While vanilla ES generates i.i.d. perturbations ϵ_i, orthogonal ES couples the perturbations such that $\epsilon_i'^T \epsilon_j' = 0, i \neq j$.

Denote d as the dimensionality of the parameter space $\mu \in \mathbb{R}^d$ and let $\rho = \sigma_2 / \sigma_1$. Let $\hat{J}(\pi_{\mu 1})$ be a one-sample noisy estimate of $J(\pi_\mu)$. Recall that ES gradient takes the form $\hat{g}_\mu^{\text{ES}} = 1/N \sum_{i=1}^{N} \hat{J}(\pi_{\mu + \sigma_1 \epsilon_i}) \, \epsilon_i / \sigma_1$ (eq. (4)). The orthogonal ES gradient takes the same form, but with orthogonal perturbations $\hat{g}_\mu^{\text{ort}} = 1/N \sum_{i=1}^{N} \hat{J}(\pi_{\mu + \sigma_1 \epsilon_i'}) \, \epsilon_i' / \sigma_1$. Finally, the ES with control variate produces an estimator of the form (8). We are now ready to provide the following theoretical result.

Theorem 1. *In the one-step* MDP *described above, the ratio of the variance of the orthogonal* ES *to the variance of the vanilla* ES, *and the corresponding ratio for the control variate* ES *satisfy respectively:*

$$\frac{\mathbb{V}[\hat{g}_\mu^{\text{ort}}]}{\mathbb{V}[\hat{g}_\mu^{\text{ES}}]} = 1 - \frac{N-1}{(1+\rho^2)d+1},$$
$$\frac{\mathbb{V}[\hat{g}_\mu^{\text{CV}}]}{\mathbb{V}[\hat{g}_\mu^{\text{ES}}]} \leq 1 - \frac{\rho^2[d((1+\rho^2)-4]}{[(1+\rho^2)d+1](1+\rho^2)}.$$

As a result, there exists a threshold ρ_0 such that when $\rho \geq \rho_0$, we always have $\mathbb{V}[\hat{g}_\mu^{\text{CV}}] \leq \mathbb{V}[\hat{g}_\mu^{\text{ort}}]$. (See: Appendix for details). Importantly, when d is large enough, we have $\rho_0 \to \sqrt{N/d}$.

Some implications of the above theorem: **(1)** For orthogonal ES, the variance reduction depends explicitly on the sample size N. In cases where N is small, the variance gain over vanilla ES is not significant. On the other hand, $\mathbb{V}[\hat{g}_\mu^{\text{CV}}]$ depends implicitly on N because in practice η^* is approximated via gradient decent and large sample size N leads to more stable updates; **(2)** The threshold ρ_0 is useful in practice. In high-dimensional applications where sample efficiency is important, we have large d and small N. This implies that for a large range of the ratio $\rho = \frac{\sigma_2}{\sigma_1} \geq \rho_0$, we could expect to achieve more variance reduction than orthogonal ES. **(3)** The above derivation is based on

the simplification of the general multi-step MDP problem. The practical performance of the control variate can also be influenced by how well η^* is estimated. Nevertheless, we expect this example to provide some guideline as to how the variance reduction property of ES with control variate depends on ρ and N, in contrast to orthogonal ES; **(4)** The theoretical guarantee for variance reduction of orthogonal ES (Choromanski et al., 2018) relies on the assumption that $\hat{J}(\pi_{\theta'})$ can be simulated without noise [1], which does not hold in practice. In fact, in RL the noise of the reward estimate heavily depends on the policy $\pi_{\theta'}$ (intuitively the more random the policy is, the more noise there is in the estimate). On the other hand, ES with control variate depends less on such assumptions but rather relies on finding the proper scalar η using gradient descent. We will see in the experiments that this latter approach reliably improves upon the ES baseline.

4 Experiments

In the experiments, we aim to address the following questions: **(1)**: Does the control variate improve downstream training through variance reduction? **(2)**: How does it compare with other recent variance reduction techniques for ES?

To address these questions, we compare our control variate to general purpose variance reduction techniques on a wide range of high-dimensional RL tasks with continuous action space.

Antithetic sampling: See Salimans et al. (2017). Let $\epsilon_i, 1 \leq i \leq N$ be the set of perturbation directions. Antithetic sampling perturbs the policy parameter with a set of antithetic pairs $(\epsilon_i, \epsilon_i'), 1 \leq i \leq N$ where $\epsilon_i' = -\epsilon_i$.

Orthogonal directions (ORTHO): See Choromanski et al. (2018). The set of perturbations $\epsilon_i, 1 \leq i \leq N$ applied to the policy parameter are generated such that they have the same marginal Gaussian distributions but are orthogonal to each other $\epsilon_i^T \epsilon_j = 0, i \neq j$.

Geometrically coupled MC sampling (GCMC): See Rowland et al. (2018). For each antithetic pair $(\epsilon_i, \epsilon_i')$, GCMC couples their length such that $F_R(\|\epsilon_i\|) + F_R(\|\epsilon_i'\|) = 1$ where F_R is the CDF of the norm of a standard Gaussian with the same dimension as ϵ_i.

Quasi Monte-Carlo (QMC): See Choromanski et al. (2018); Rowland et al. (2018). QMC first

---

[1]The variance reduction proof can be extended to cases where $\hat{J}(\pi_{\theta'})$ has the same level of independent noise for all θ'.

generates a low-discrepancy Halton sequence $\{r_i\}_{i=1}^N$ in $[0,1]^d$ with N elements where d is the dimension of parameter θ. Then apply the inverse CDF F_g^{-1} of a standard univariate Gaussian elementwise to the sequence $\epsilon_i = F_g^{-1}(r_i)$ to generate perturbation vectors.

We find that our RL-specific control variate achieves outperforms these general purpose variance reduction techniques. In addition to the above baseline comparison, we also compare CV with more advanced policy gradient algorithms such as TRPO / PPO (Schulman et al., 2015b, 2017), as well as deterministic policy + ES baseline (Mania et al., 2018).

Implementation details. Since antithetic sampling is the most commonly applied variance reduction method, we combine it with the control variate, OR-THO, GCMC and QMC. The policy π_θ is parameterized as a neural network with 2 hidden layers each with 32 units and relu activation function. The output is a vector $\mu_\theta \in \mathbb{R}^K$ used as a Gaussian mean, with a separately parameterized diagonal vector $\sigma_0 \in \mathbb{R}^K$ independent of state. The action is sampled the Gaussian $a \sim \mathcal{N}(\mu_\theta(s), \text{diag}(\sigma^2))$. The backpropagation pipeline is implemented with Chainer (Tokui et al., 2015). The learning rate is $\alpha = 0.01$ with Adam optimizer (Kingma and Ba, 2014), the perturbation standard deviation $\sigma = 0.02$. At each iteration we have $N = 5$ distinct perturbations ϵ_i ($2N$ samples in total due to antithetc sampling). For the control variate (8), the discount factor is initialized to be $\gamma = 0.99$ and updated with ES, we introduce the details in the Appendix. The control variate scaling factor η is updated with learning rate selected from $\alpha_\eta \in \{10^{-3}, 10^{-4}, 10^{-5}\}$. As commonly practiced in prior works (Salimans et al., 2017; Mania et al., 2018; Choromanski et al., 2018), in order to make the gradient updates less sensitive to the reward scale, returns are normalized before used for computing the gradients. We adopt this technique and discuss the details in the Appendix. Importantly, we do not normalize the observations (as explained in (Mania et al., 2018)) to avoid additional biasing of the gradients.

Benchmark tasks and baselines. To evaluate how variance reduction impacts downstream policy optimization, we train neural network policies over a wide range of high-dimensional continuous control tasks, taken from OpenAI gym (Brockman et al., 2016), Roboschool (Schulman et al., 2017) and DeepMind Control Suites (Tassa et al., 2018). We introduce their details below. We also include a LQR task suggested in (Mania et al., 2018) to test the stability of the gradient update for long horizons ($T = 2000$). Details of the tasks are in the Appendix. The policies are trained with five variance reduction settings: Vanilla ES baseline; ES with orthogonalization (ORTHO); ES with GCMC

(GCMC); ES with Quasi-MC (QMC); and finally our proposed ES with control variate (CV).

Results. In each subplot of Figure 2, we present the learning curves of each variance reduction technique, with average performance over 5 random seeds and the shaded areas indicate standard deviations.

We make several observations regarding each variance reduction technique: (1) Though ORTHO and GCMC significantly improve the learning progress over the baseline ES under certain settings (e.g. ORTHO in Swimmer and GCMC), their improvement is not very consistent. In certain cases, adding such techniques even makes the performance worse than the baseline ES. We speculate that this is because the variance reduction conditions required by these methods are not satisfied, e.g. the assumption of noiseless estimate of returns. Overall, ORTHO is more stable than GCMC and QMC; (2) QMC performs poorly on most tasks. We note that similar results have been reported in (Rowland et al., 2018) where they train a navigation policy using QMC and show that the agent does not learn at all. We speculate that this is because the variance reduction achieved by QMC is not worth the bias in the RL contexts; (3) No variance reduction technique performs uniformly best across all tasks. However, CV performs the most consistently and achieves stable gains over the vanilla ES baseline, while other methods can underperform the vanilla ES baseline.

To make clear the comparison of final performance, we record the final performance (mean ± std) of all methods in Table 1. Best results across each task are highlighted in bold font. For a fair presentation of the results, in cases where multiple methods achieve statistically similar performance, we highlight all such methods. CV consistently achieves the top results across all reported tasks.

Comparison with Policy Gradients. A natural question is what happens when we update the policy just based on the PG estimator? We show the complete comparison in Table 1, where we find pure PG to be mostly outperformed by the other ES baselines. We speculate that this is because the vanilla PG are themselves quite unstable, as commonly observed in prior works on PG which aim to alleviate the instability by introducing bias in exchange for smaller variance (Schulman et al., 2015b; Mnih et al., 2016; Schulman et al., 2017). We provide a detailed review in the Appendix. This comparison also implies that a careful control variate scheme can extract the benefit of PG estimators for variance reduction in ES, instead of completely relying on PG.

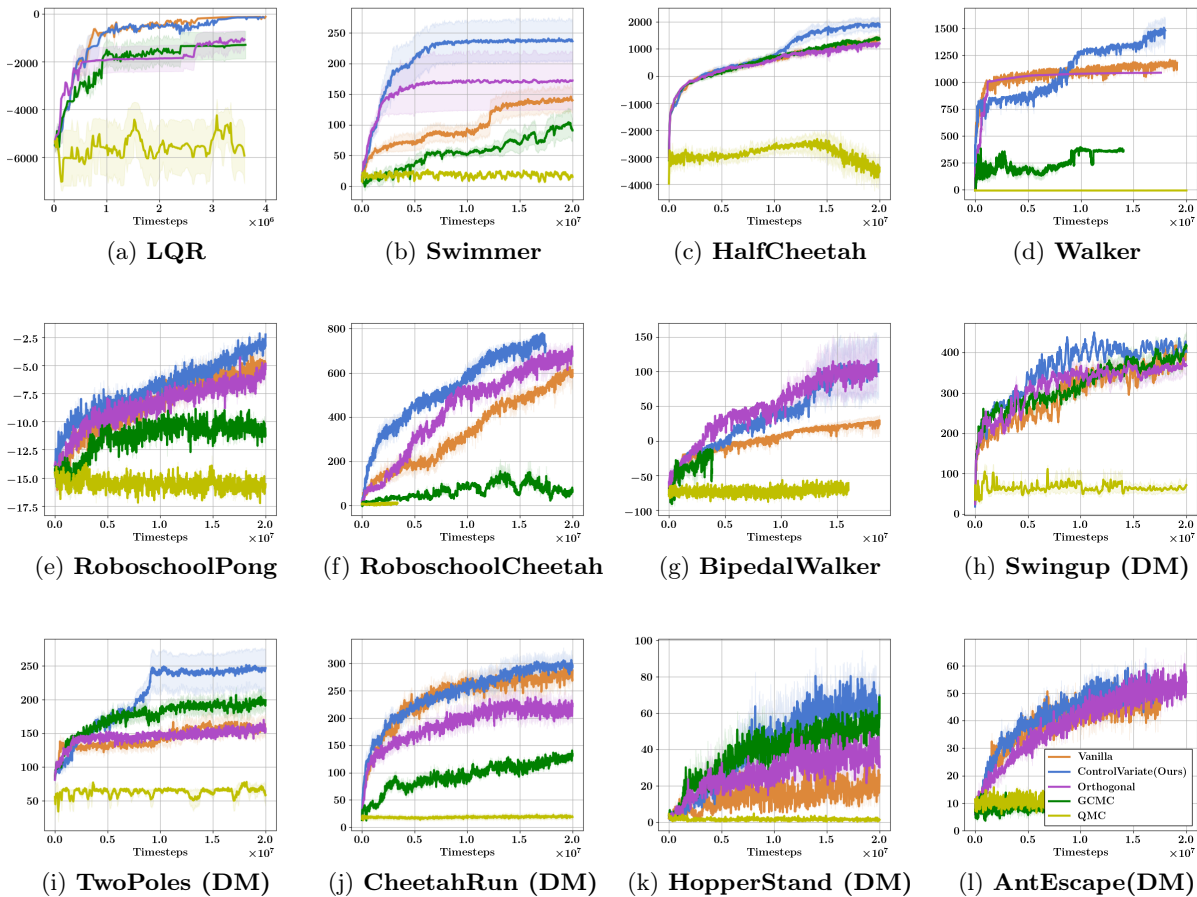To assess the impact of our method, we also com-

Figure 2: Training performance on Continuous Control Benchmarks: Swimmer, HalfCheetah, CartPole + {Swingup, TwoPoles, Balance}, Pendulum Swingup, Cheetah Run and Hopper. Tasks with DM stickers are from the DeepMind Control Suites. We compare five alternatives: baseline ES (orange), CV (blue, ours), ORTHO (marron), GCMC (green) and QMC (yellow). Each task is trained for $2 \cdot 10^7$ time steps (LQR is trained for $4 \cdot 10^6$ steps) and the training curves show the mean $\pm$ std cumulative rewards across 5 random seeds.

pare with trust-region based on-policy PG algorithms: TRPO (Schulman et al., 2015b) and PPO (Schulman et al., 2017). Instead of a batched implementation as in the original paper, we adopt a fully on-line PG updates to align with the way ES baselines are implemented. We list detailed hyper-parameters and implementations in the Appendix C. Due to space limit, the results of selected tasks are presented in table 2 and we leave a more comprehensive set of results in the Appendix.

We see that the trust-region variants lead to generally more stable performance than the vanilla PG. Nevertheless, the performance of these algorithms do not match those of the ES baselines. There are several enhancements one can make to improve these PG baselines, all generally relying on biasing the gradients in exchange for smaller variance, e.g. normalizing the observations and considering a biased objective (Dhariwal et al., 2017). While here we consider only the 'unbiased' PG

methods, we discuss these methods in the Appendix.

Comparison with ES baselines. As mentioned in Section 3, the CV is applicable only in cases where policies are stochastic. However, prior works consider mainly deterministic policy for continuous control (Mania et al., 2018; Choromanski et al., 2018). To assess the effects of policy choices, we adopt the same ES baseline pipeline but with deterministic policies. The results of deterministic policies are reported in Table 2. We make several observations: (1) Deterministic policies generally perform better than stochastic policies when using ES baselines. We speculate this is because the additional noise introduced by the stochastic policy is not worth the additional variance in the gradient estimation. Nevertheless, CV can leverage the variance reduction effects thanks to the stochastic policy, and maintain generally the best performance across tasks. Such comparison illustrates that it is beneficial

Table 1: Final performance on benchmark tasks. The policy is trained for a fixed number of steps on each task. The result is mean ± std across 5 random seeds. The best results are highlighted in bold font. We highlight multiple methods if their results cannot be separated (mean ± std overlap). CV (ours) achieves consistent gains over the baseline and other variance reduction methods. We also include a PG baseline.

Tasks	Vanilla ES	Orthogonal	GCMC	QMC	CV (Ours)	PG
LQR	-176 ± 12	-1337 ± 573	-1246 ± 502	-5634 ± 1059	$\mathbf{-143 \pm 4}$	-7243 ± 275
SWIMMER	141 ± 20	171 ± 47	94 ± 19	16 ± 2	$\mathbf{237 \pm 33}$	-132 ± 5
HALFCHEETAH	1339 ± 178	1185 ± 76	1375 ± 58	-3466 ± 338	$\mathbf{1897 \pm 232}$	-180 ± 4
WALKER	1155 ± 34	1087 ± 1	360 ± 4	6 ± 0	$\mathbf{1476 \pm 112}$	282 ± 25
PONG(R)	-5.0 ± 0.8	-5.5 ± 0.3	-10.6 ± 0.4	-15.6 ± 0.3	$\mathbf{-3.0 \pm 0.3}$	-17 ± 0.2
HALFCHEETAH(R)	$\mathbf{595 \pm 42}$	$\mathbf{685 \pm 34}$	68 ± 8	11 ± 2	$\mathbf{709 \pm 16}$	12 ± 0
BIPEDALWALKER	25 ± 9	$\mathbf{107 \pm 31}$	-19 ± 5	-70 ± 3	$\mathbf{105 \pm 40}$	-82 ± 12
CHEETAH(DM)	281 ± 15	217 ± 15	129 ± 4	18 ± 5	$\mathbf{296 \pm 15}$	25 ± 6
PENDULUM(DM)	20 ± 3	$\mathbf{54 \pm 17}$	25 ± 8	11 ± 2	$\mathbf{43 \pm 1}$	3 ± 1
TWOPOLES(DM)	159 ± 13	158 ± 2	196 ± 12	62 ± 12	$\mathbf{245 \pm 29}$	14 ± 1
SWINGUP(DM)	$\mathbf{394 \pm 15}$	$\mathbf{369 \pm 22}$	414 ± 31	67 ± 14	$\mathbf{406 \pm 26}$	55 ± 10
BALANCE(DM)	692 ± 57	771 ± 41	$\mathbf{995 \pm 1}$	223 ± 32	847 ± 71	401 ± 12
HOPPERHOP(DM)	$\mathbf{5.7 \pm 2.1}$	$\mathbf{6.8 \pm 0.7}$	0.3 ± 0.1	0.0 ± 0.0	$\mathbf{6.5 \pm 1.5}$	0.1 ± 0.0
STAND(DM)	21 ± 5	36 ± 10	$\mathbf{54 \pm 4}$	1.0 ± 0.2	$\mathbf{60 \pm 11}$	0.5 ± 0.1
ANTWALK(DM)	$\mathbf{200 \pm 19}$	$\mathbf{234 \pm 10}$	82 ± 11	133 ± 9	$\mathbf{239 \pm 10}$	100 ± 11
ANTESCAPE(DM)	$\mathbf{47 \pm 3}$	$\mathbf{52 \pm 3}$	8 ± 2	10 ± 1	$\mathbf{51 \pm 2}$	6 ± 1

Table 2: Final performance on benchmark tasks. The setup is the same as in Table 2. CV (ours) achieves consistent gains over deterministic policy + ES as well as more advanced baselines such as TRPO and PPO. We also include a PG baseline for easy comparison. In the following, 'Det' denotes the deterministic policies.

Tasks	PPO	TRPO	Det	CV (Ours)	PG
SWIMMER	11 ± 7	11 ± 7	191 ± 100	$\mathbf{237 \pm 33}$	-132 ± 5
HALFCHEETAH	-175 ± 20	-174 ± 16	1645 ± 1298	$\mathbf{1897 \pm 232}$	-180 ± 4
WALKER	657 ± 291	657 ± 292	1588 ± 744	$\mathbf{1476 \pm 112}$	282 ± 25
PONG(R)	-17.1 ± 0.4	-15.0 ± 2.6	-10.4 ± 5.4	$\mathbf{-3.0 \pm 0.3}$	-17 ± 0.2
HALFCHEETAH(R)	14 ± 2	13 ± 3	502 ± 199	$\mathbf{709 \pm 16}$	12 ± 0
BIPEDALWALKER	-66 ± 39	-66 ± 38	1 ± 2	$\mathbf{105 \pm 40}$	-82 ± 12
PENDULUM(DM)	0.6 ± 0.5	7.5 ± 4.8	40 ± 16	$\mathbf{43 \pm 1}$	3 ± 1
BALANCE(DM)	264 ± 46	515 ± 42	692 ± 250	$\mathbf{847 \pm 71}$	401 ± 12
HOPPERHOP(DM)	0.2 ± 0.2	0.0 ± 0.0	4.6 ± 6.2	$\mathbf{6.5 \pm 1.5}$	0.1 ± 0.0
ANTWALK(DM)	96 ± 36	180 ± 41	192 ± 20	$\mathbf{239 \pm 10}$	100 ± 11

to combine stochastic policies with ES methods as long as there is proper variance reduction.

Practical Scalability. In practical applications, the scalability of algorithms over large computational architecture is critical. Since CV blends ideas from ES with PG, we require both distributed sample collection (Salimans et al., 2017; Mania et al., 2018) and distributed gradient computation (Dhariwal et al., 2017). Since both components can be optimally implemented over large architectures, we expect CV to properly scale. A more detailed discussion is in the Appendix.

5 Conclusion

We constructed a control variate for ES that take advantage of the MDP structure of RL problems to improve on

state-of-the-art variance reduction methods for ES algorithms. Training algorithms using our control variate outperform those applying general-purpose MC methods for variance reduction. We provided theoretical insight into the effectiveness of our algorithm as well as exhaustive comparison of its performance with other methods on the set of over 16 RL benchmark tasks. In principle, this control variate can be combined with other variance reduction techniques; this may lead to further performance gains.

We leave as future work to study how similar structured control variates can be applied to improve the performance of state-of-the-art PG algorithms, in particular, cases where gradients have already been deliberately biased to achieve better performance such as in (Dhariwal et al., 2017).

References

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. (2016). Openai gym. *arXiv preprint arXiv:1606.01540*.

Choromanski, K., Rowland, M., Sindhwani, V., Turner, R., and Weller, A. (2018). Structured evolution with compact architectures for scalable policy optimization. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 970–978, Stockholmsmässan, Stockholm Sweden. PMLR.

Dhariwal, P., Hesse, C., Klimov, O., Nichol, A., Plappert, M., Radford, A., Schulman, J., Sidor, S., and Wu, Y. (2017). Openai baselines. `https://github.com/openai/baselines`.

Grathwohl, W., Choi, D., Wu, Y., Roeder, G., and Duvenaud, D. (2018). Backpropagation through the void: Optimizing control variates for black-box gradient estimation. In *International Conference on Learning Representations*.

Gu, S., Levine, S., Sutskever, I., and Mnih, A. (2016). Muprop: Unbiased backpropagation for stochastic neural networks. In *4th International Conference on Learning Representations (ICLR)*.

Gu, S., Lillicrap, T., Ghahramani, Z., Turner, R. E., and Levine, S. (2017). Q-prop: Sample-efficient policy gradient with an off-policy critic. In *Proceedings International Conference on Learning Representations (ICLR)*. OpenReviews.net.

Jang, E., Gu, S., and Poole, B. (2017). Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In *International Conference on Learning Representations*.

Leordeanu, M. and Hebert, M. (2008). Smoothing-based optimization. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE.

Liu, H., Feng, Y., Mao, Y., Zhou, D., Peng, J., and Liu, Q. (2017). Action-depedent control variates for policy optimization via stein's identity. *arXiv preprint arXiv:1710.11198*.

Maddison, C. J., Mnih, A., and Teh, Y. W. (2017). The concrete distribution: a continuous relaxation of discrete random variables. In *International Conference on Learning Representations*.

Mania, H., Guy, A., and Recht, B. (2018). Simple random search of static linear policies is competitive for reinforcement learning. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31*, pages 1800–1809. Curran Associates, Inc.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1928–1937.

Paisley, J. W., Blei, D. M., and Jordan, M. I. (2012). Variational bayesian inference with stochastic search. In *ICML*. icml.cc / Omnipress.

Ranganath, R., Gerrish, S., and Blei, D. (2014). Black box variational inference. In *Artificial Intelligence and Statistics*, pages 814–822.

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics*.

Ross, S. (2002). Simulation, 2002.

Rowland, M., Choromanski, K. M., Chalus, F., Pacchiano, A., Sarlos, T., Turner, R. E., and Weller, A. (2018). Geometrically coupled monte carlo sampling. In *Advances in Neural Information Processing Systems*, pages 195–206.

Salimans, T., Ho, J., Chen, X., Sidor, S., and Sutskever, I. (2017). Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*.

Schulman, J., Heess, N., Weber, T., and Abbeel, P. (2015a). Gradient estimation using stochastic computation graphs. In *Advances in Neural Information Processing Systems*, pages 3528–3536.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. (2015b). Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897.

Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P. (2016). High-dimensional continuous control using generalized advantage estimation. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. (2014). Deterministic policy gradient algorithms. In *ICML*.

Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063.

Tassa, Y., Doron, Y., Muldal, A., Erez, T., Li, Y., Casas, D. d. L., Budden, D., Abdolmaleki, A., Merel, J., Lefrancq, A., et al. (2018). Deepmind control suite. *arXiv preprint arXiv:1801.00690*.

Tokui, S., Oono, K., Hido, S., and Clayton, J. (2015). Chainer: a next-generation open source framework for deep learning. In *Proceedings of workshop on machine learning systems (LearningSys) in the twenty-ninth annual conference on neural information processing systems (NIPS)*, volume 5, pages 1–6.

Tucker, G., Bhupatiraju, S., Gu, S., Turner, R., Ghahramani, Z., and Levine, S. (2018). The mirage of action-dependent baselines in reinforcement learning. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5015–5024, Stockholmsmässan, Stockholm Sweden. PMLR.

Tucker, G., Mnih, A., Maddison, C. J., Lawson, J., and Sohl-Dickstein, J. (2017). Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models. In *Advances in Neural Information Processing Systems*, pages 2627–2636.

Wu, C., Rajeswaran, A., Duan, Y., Kumar, V., Bayen, A. M., Kakade, S., Mordatch, I., and Abbeel, P. (2018). Variance reduction for policy gradient with action-dependent factorized baselines. In *International Conference on Learning Representations*.