Dimensionality Reduction and Prioritized Exploration for Policy Search

Anonymous Author
Anonymous Institution

Abstract

Black-box policy optimization, a class of reinforcement learning algorithms, explore and update policies at the parameter level. These algorithms are applied widely in robotics applications with movement primitives and non-differentiable policies. These methods are particularly relevant where exploration at the action level could lead to actuator damage or other safety issues. However, this class of algorithms does not scale well with the increasing dimensionality of the policy, leading to high demand for samples that are expensive to obtain on real-world systems. In most systems, policy parameters do not contribute equally to the return. Thus, identifying those parameters which contribute most allows us to narrow the exploration and speed up learning. Updating only the effective parameters requires fewer samples, solving the scalability issue. We present a novel method to prioritize exploration of effective parameters, coping with full covariance matrix updates. Our algorithm learns faster than recent approaches and requires fewer samples to achieve state-of-the-art results. To select these effective parameters, we consider both the Pearson correlation coefficient and the Mutual Information. We showcase the capabilities of our approach using the Relative Entropy Policy Search algorithm in several simulated environments, including robotics simulations.

Preliminary work. Under review by AISTATS 2022. Do not distribute.

1 INTRODUCTION

Black-Box Optimization (BBO) is a class of Policy Search methods that tackle Reinforcement Learning (RL) problems in the parameter space. These methods have been widely applied to robotics applications, such as Table Tennis [Peters et al., 2010], Beer Pong [Abdolmaleki et al., 2015], Pancake-Flipping [Kormushev et al., 2010], and Juggling [Ploeger et al., 2020]. By using a search distribution, it conducts the exploration at the parameter level instead of exploring at the action level. Exploring at the parameter level not only leads to more reliable policy updates but is also better suited for real-world learning tasks, since action level exploration may damage the actuator or is low-pass filtered by the physical system [Deisenroth et al., 2013].

One major challenge in BBO is the scalability to high-dimensional tasks. The search distribution of the policy parameters is often represented as a multivariate Gaussian distribution. While the search distribution parameterized by a diagonal covariance matrix is sufficient for simple tasks, more complex tasks require correlated exploration using a full covariance matrix to increase the learning speed. However, the dimensionality of the full covariance matrix scales quadratically over the dimensionality of the parameters. Therefore, the learning agent requires more samples to update the search distribution properly. In fact, particularly when samples come directly from real robots, exploration is problematic in terms of time and resource costs.

To reach the optimal policy with fewer samples, we usually tackle the problem in two aspects: more informative samples or more effective policy updates. Noticeably, the policy parameters in a system generally do not have equal contribution to the return. For instance, many tasks using a 7 Degrees of Freedom (DoF) manipulator can be solved without utilizing all DoFs. We refer to the parameters that contribute most to the return as *effective parameters* and the remaining ones as *ineffective parameters*. Samples spreading over the *ineffective parameters* do not provide valu-

able information to the learning agent, and updating the distribution w.r.t such parameters does not have an impact on the learning performance. On the contrary, if the sampling and the policy update focus on the *effective parameters*, we could obtain better learning performance.

In this paper, we propose a new BBO algorithm, Dimensionality Reduced Constrained REPS (DR-CREPS). Our method consists of 3 components: i. *Parameter Effectiveness Metric*. To estimate the effectiveness of the parameter, we estimate the correlation measurement (e.g., Pearson Correlation Coefficient (PCC), Mutual Information (MI)) between parameters and the episodic return. ii. *Prioritized Exploration*. During the exploration process, we concentrate the sampling on the *effective* parameters by decreasing the covariance entries w.r.t the *ineffective* parameters. iii. *Guided Dimensionality Reduction*. We propose a dimensionality reduction technique for policy search algorithms that scales to the full covariance case by partially updating the search distribution w.r.t. the effective parameters. For simplicity, we first present i. for the diagonal covariance matrix case. Then, we project the full covariance matrix into a rotated space with diagonal covariance matrix. Subsequently, we explain ii. and iii. in the rotated space. We update the policy in the rotated space following the Constrained REPS (CREPS) algorithm [Ploeger et al., 2020]. Finally, we evaluate the proposed methodology in four continuous control and simulated robotics tasks. The empirical results show the benefit of considering effective parameters for BBO.

Assumptions and Limitations

In this work, we focus on learning the simple parametric policies, such as Movement Primitives and Linear Feedback Controller, under a multivariate Gaussian search distribution. While these policies can be high dimensional in the parameter space, each parameter may have a great impact on the policies behavior. This class of policies includes many Movement Primitives used in robotics, linear policies, and ad-hoc, non-differentiable policies, but excludes the neural network scenario, where each parameter has no major impact on the policy. Thus, extending the dimensionality reduction techniques presented in this work to the neural network scenario is nontrivial and requires further investigation. One possible solution in the literature is to consider specific neural network structures [Choromanski et al., 2018].

Parameter exploration and BBO approaches might be well suited for a wide variety of tasks, e.g., when the reward function is not much informative. However, classical exploration could potentially be more sample efficient in environments where the local policy changes affect the total return, e.g., standard Mujoco environments. For this reason, we focus on specific robotics tasks where the task reward is sparse and the black box learning could be beneficial.

Related Work

Dimensionality reduction is an important approach to alleviate the problem of "Curse of Dimensionality". It has been studied from various perspectives in Imitation Learning (IL) and RL, e.g., state space [Bitzer et al., 2010, Jaderberg et al., 2017], action space [Luck et al., 2014] and parameter space [Bitzer and Vijayakumar, 2009, Colomé et al., 2014]. In this paper, we will focus on the dimensionality reduction problem in the parameter space.

In [Ben Amor et al., 2012], the authors tackle this problem by directly projecting the movement primitives into a lower-dimensional space. In their method, they use the Principal Component Analysis (PCA) and show its application to a human grasping task. Colomé and Torras extensively studied the PCA approach in different types of Movement Primitives, such as Dynamic Movement Primitives (DMP) [Colomé and Torras, 2014, Colomé and Torras, 2018a] and Probabilistic Movement Primitives (ProMP) [Colomé et al., 2014, Colomé and Torras, 2018b]. The authors showcase the method on trajectory learning with movement primitives for various applications including bimanual manipulation of clothes. A different approach to dimensionality reduction for movement primitives is the reduction of the parameter space via a Mixture of Probabilistic Principal Component Analysis (MPPCA) by [Tosatto et al., 2021] which makes the optimization feasible for the RL setting. In [Bitzer and Vijayakumar, 2009], the authors propose Gaussian Process Latent Variable Model (GPLVM) as a non-linear dimensionality reduction technique and apply it to high dimensional DMPs. [Delgado-Guerrero et al., 2020] also utilize a GPLVM to project the data itself into a lower-dimensional space. They build a surrogate model of the reward function which they incentivize by weighing the data according to the MI with the reward.

In [Ewerton et al., 2019] the authors exploit the connection of the PCC with the relevancy of parameters in trajectories to different objectives. They show that the computed PCC can improve the policy optimization of trajectory distributions by reducing the exploration space. To go beyond the PCC for dimensionality reduction, we also evaluate the MI as a non-linear correlation measure. The MI captures the informa-

tion overlap between two random variables and allows us to measure how parameters influence the total return of an episode. Since its computation from samples is nontrivial, we turn to [Carrara and Ernst, 2019, Kraskov et al., 2004] who provide suitable estimators. Another drawback of [Ewerton et al., 2019] is the limitation to diagonal covariance matrices which we overcome using Singular Value Decomposition (SVD) and the importance estimation of the parameters.

2 PROPOSED METHOD

Our method consists of three main components: First, we select the effective parameters using a correlation metric of choice. Second, we sample a new parameter vector by prioritizing exploration on the effective parameters. Third, we update the search distribution by prioritizing the important policy parameters. We provide a step-wise visualization of our methods applied to a full covariance matrix in Fig. 1.

2.1 Preliminaries

A Markov Decision Process (MDP) is defined as a 6-tuple $\mathcal{M} =< \mathcal{X}, \mathcal{U}, \mathcal{P}, r, \iota, \gamma >$, where $\mathcal{X}$ is the state space, $\mathcal{U}$ is the action space, $\mathcal{P} : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ is the transition kernel, $r : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$ is the reward function, $\iota : \mathcal{X} \to \mathbb{R}$ is the initial state distribution, and $\gamma \in (0, 1]$ is the discount factor.

A parametric policy π_θ, with $\boldsymbol{\theta} \in \mathbb{R}^n$ provides the action $\boldsymbol{u} \in \mathcal{U}$ to perform in each state $\boldsymbol{x} \in \mathcal{X}$. The performance of a policy, given a parameter vector $\boldsymbol{\theta}$, is evaluated through the return, i.e., the discounted cumulative reward:

$$J(\pi_\theta) = \mathop{\mathbb{E}}_{(\boldsymbol{x}_t, \boldsymbol{u}_t) \sim \pi_\theta, \mathcal{P}, \iota} \left[\sum_{t=0}^{T} \gamma^t r(\boldsymbol{x}_t, \boldsymbol{u}_t) \right],$$

where T is the length of the trajectory. Different to standard step-based RL, in the BBO setting we try to maximize the expected return $\mathcal{J}$ under a search distribution p:

$$\mathcal{J}(p) = \mathbb{E}_{\boldsymbol{\theta} \sim p} \left[J(\pi_\theta) \right].$$

In this paper, we focus on Gaussian search distributions. We define the distribution at epoch k as

$$p_k(\boldsymbol{\theta}) = \mathcal{N}\left(\cdot | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k \right),$$

with mean $\boldsymbol{\mu}_k \in \mathbb{R}^n$, and covariance $\boldsymbol{\Sigma}_k \in \mathbb{R}^{n \times n}$. To be a proper probability distribution, $\boldsymbol{\Sigma}_k$ must be positive definite.

A typical BBO approach explores the environment by sampling $\boldsymbol{\theta}$ from the search distribution p and evaluating the performance of the policy parameters by interacting with the environment. After the evaluation of N parameters, the algorithm updates the search distribution using the parameters matrix $\boldsymbol{\Theta} \in \mathbb{R}^{N \times n}$, which contains all the sampled parameter vectors, and the vector of returns $J(\boldsymbol{\Theta}) \in \mathbb{R}^N$, containing the corresponding performance of each parameter vector. In the rest of the paper, we denote the j-th component of i-th parameter sample $\boldsymbol{\theta}_i$ as θ_i^j.

To improve the robustness of the learning behavior and avoid premature convergence, many RL algorithms, such as [Peters et al., 2010, Schulman et al., 2015], apply the Kullback-Leibler Divergence (KL) constraint to the policy update. The resulting BBO optimization problem is

$$\max_p \ \mathcal{J}(p), \quad \text{s.t. } \mathrm{KL}\left[p_k \| p_{k-1} \right] \leq \epsilon. \tag{1}$$

Using the method of Lagrangian multipliers, Relative Entropy Policy Search (REPS) calculates the solution for this problem as $p_k(\boldsymbol{\theta}) \propto p_{k-1}(\boldsymbol{\theta}) \exp(J(\pi_\theta)/\eta)$, with the temperature parameter η automatically chosen to fulfill the KL constraint. As the proposed update is intractable, the authors approximate the new policy p_k using a Gaussian $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$, through sample-based Weighted Maximum Likelihood Estimate (WMLE). However, this results in a distribution that is no longer guaranteed to fulfill the original KL constraint. CREPS, introduced in [Ploeger et al., 2020], addresses this problem by adding the KL constraint to the WMLE fit. The Constrained WMLE (CWMLE) update is formulated as:

$$\begin{aligned} &\max_{p_{k+1}} \sum_{i=1}^{N} d_i \log p_{k+1}\left(\boldsymbol{\theta}_i \right) \\ &\text{s.t. } \mathrm{KL}\left(p_k || p_{k+1} \right) \leq \epsilon, \quad \mathrm{H}\left(p_k \right) - \kappa \leq \mathrm{H}\left(p_{k+1} \right), \end{aligned} \tag{2}$$

with the search distribution p_k at epoch k, the weight d_i corresponding to the parameter sample $\boldsymbol{\theta}_i$, the KL bound ϵ, and the maximal allowed entropy decrease κ.

2.2 Estimating Effective Parameters

To determine the *effective parameters* and conversely the *ineffective parameters*, we need a Parameter Effectiveness Metric to measure the influence of the policy parameters on the episodic return. Several metrics are proposed in the pioneering work, such as Pearson Correlation Coefficient for policy updates [Ewerton et al., 2019] and Mutual Information (referred to as PIC [Furuta et al., 2021]) for the task complexity evaluation.

Pearson Correlation Coefficient measures the linear relationship between two random variables which can be directly estimated based on samples. The coefficient is bound by $[+1, -1]$ for a fully linear positive and negative correlation respectively. We measure the

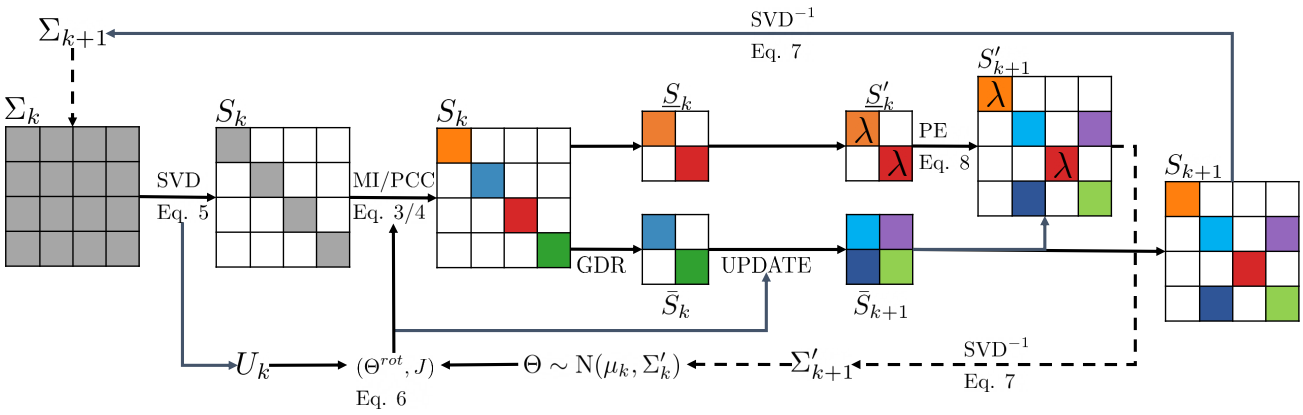


Figure 1: Guided Dimensionality Reduction (GDR) and prioritized exploration (PE) on a full covariance update for a generic policy search algorithm (UPDATE).

PCC of the j-th component $\boldsymbol{\Theta}^j$ with the cumulative return $\boldsymbol{J}$ based on the samples and use the absolute value as the correlation measure

$$C_{PCC}[\boldsymbol{\Theta}^j; \boldsymbol{J}] = \frac{\left|\sum_i \left(\theta_i^j - \hat{\theta}\right)\left(J_i - \hat{J}\right)\right|}{\sqrt{\sum_i \left(\theta_i^j - \hat{\theta}^j\right)^2 \sum_i \left(J_i - \hat{J}\right)^2}}, \quad (3)$$

where $\hat{\theta}^j$ and $\hat{J}$ are the respective means.

Mutual Information on the other hand possesses an attractive information-theoretic interpretation and can be used to measure non-linear correlations. In theory, MI is able to capture a more complex relationship between two random variables than PCC. On the flipside, estimating it directly from samples is difficult, since it is defined in terms of probability densities over the random variables which require a density estimation from samples. The MI between the j-th policy parameter and the return is

$$C_{MI}[\boldsymbol{\Theta}^j; \boldsymbol{J}] = \int p\left(\boldsymbol{\theta}^j, \boldsymbol{J}\right) \log \frac{p\left(\boldsymbol{\theta}^j, \boldsymbol{J}\right)}{p\left(\boldsymbol{\theta}^j\right) p\left(\boldsymbol{J}\right)} d\boldsymbol{\theta} d\boldsymbol{J}. \quad (4)$$

After measuring the correlation between the parameters and the return, we select a fixed number of parameters with the highest correlation and assign them to the effective class $\bar{\varepsilon}$. The remaining parameters are considered as the ineffective class $\underline{\varepsilon}$.

2.3 The Full Covariance Case

As presented by Ewerton et al. [Ewerton et al., 2019], finding the corresponding effectiveness or relevance of the parameters is trivial for the diagonal covariance case. However, this approach neglects the inter-dimensional dependencies and leads to limitations in the policy complexity and the learning efficiency. We, therefore, consider the full covariance case and propose an algorithm to overcome these limitations.

Identifying the effectiveness of the parameters in the full covariance matrix becomes nontrivial due to the relationships between them. To achieve independence among the parameters, we transform the covariance matrix into a space where every dimension is uncorrelated. This can be achieved using the SVD, which decomposes the covariance matrix into two rotation matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ as well as a diagonal matrix $\boldsymbol{S}$ with the singular values on the diagonal. More formally, decomposing covariance matrix $\boldsymbol{\Sigma}_k$ gives us

$$\boldsymbol{\Sigma}_k = \boldsymbol{U}_k \boldsymbol{S}_k \boldsymbol{V}_k^T, \quad (5)$$

with $\boldsymbol{U}_k, \boldsymbol{S}_k, \boldsymbol{V}_k \in \mathbb{R}^{n \times n}$. $\boldsymbol{\Sigma}_k$ is a positive definite square matrix, therefore we have $\boldsymbol{U}_k = \boldsymbol{V}_k$.

We then define a policy in the rotated space as

$$p_k^{rot} = \mathcal{N}\left(\cdot | \boldsymbol{\mu}_k^{\mathrm{rot}}, \boldsymbol{S}_k\right),$$

with $\boldsymbol{\mu}_k^{\mathrm{rot}} = \mathbf{0} \in \mathbb{R}^n$, $\boldsymbol{S}_k = \mathrm{diag}\left(\sigma_0^2, ..., \sigma_n^2\right) \in \mathbb{R}^{n \times n}$. Since the parameters are no longer correlated in the rotated space, it again becomes trivial to identify the covariance entries corresponding to the effective parameters. To estimate the contribution of the parameters to the total return, we must also rotated the parameter samples $\boldsymbol{\Theta}$ by

$$\boldsymbol{\Theta}^{rot} = \left(\boldsymbol{\Theta} - \boldsymbol{\mu}_k^T\right) \boldsymbol{U}_k, \quad (6)$$

which projects them into the rotated space of $\boldsymbol{S}_k$. We use the projected samples to compute the correlation between the diagonal of $\boldsymbol{S}_k$ and $\boldsymbol{J}(\Theta)$.

The introduced transformations allow us to work on a rotated policy parameterized by a diagonal covariance matrix. Note that we have to recompute the SVD at every epoch since updating $\boldsymbol{S}_k$ (in Sec. 2.5) can lead to a non-diagonal structure. Through the re-computation, we guarantee a diagonal structure of the covariance matrix before every modification or update of it. Assuming an update of $(\boldsymbol{\mu}_k^{rot}, \boldsymbol{S}_k)$ returning $(\boldsymbol{\mu}_{k+1}^{rot}, \boldsymbol{S}_{k+1})$, we must project the rotated policy p_{k+1}^{rot}

back to the original space for the policy evaluations. We utilize the previously computed $\boldsymbol{U}_k$ to project back the updated mean and the updated covariance matrix

$$\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k + \boldsymbol{U}_k \boldsymbol{\mu}_{k+1}^{\text{rot}}, \quad \boldsymbol{\Sigma}_{k+1} = \boldsymbol{U}_k \boldsymbol{S}_{k+1} \boldsymbol{U}_k^T, \quad (7)$$

yielding our new policy p_{k+1}.

2.4 Prioritized Exploration

In the following paragraph, we describe our Prioritized Exploration (PE) approach. Having identified the contribution to the total return of each parameter in the rotated space, we split them up into sets of effective $\bar{\boldsymbol{\varepsilon}}$ and ineffective parameters $\underline{\boldsymbol{\varepsilon}}$ with fixed size each. Both vectors $\bar{\varepsilon}, \underline{\varepsilon}$ capture the effectiveness of the parameters in the rotated space, i.e., similarity between $\boldsymbol{\Theta}^{\text{rot}}$ and $\boldsymbol{J}(\Theta)$ corresponding to the elements on the diagonal of $\boldsymbol{S}$. Without loss of generality, we decompose the total diagonal covariance matrix $\boldsymbol{S}$ into covariance matrices w.r.t. effective parameters and ineffective parameters denoted as $\bar{\boldsymbol{S}} \in \mathbb{R}^{m \times m}, \underline{\boldsymbol{S}} \in \mathbb{R}^{(n-m)\times(n-m)}$, respectively. We use this separation to prioritize the exploration on the effective parameters by reducing exploration in the ineffective ones. Therefore, we scale the ineffective covariance $\underline{\boldsymbol{S}}$ by hyperparameter $\lambda \in (0, 1)$ during the sampling process described as

$$\underline{\boldsymbol{S}}' = \lambda \underline{\boldsymbol{S}}. \quad (8)$$

Then, we substitute the covariance matrix corresponding to the ineffective parameter $\underline{\boldsymbol{S}}$ by the modified one $\underline{\boldsymbol{S}}'$, i.e. $\boldsymbol{S}' = \text{subst}(\boldsymbol{S}, \underline{\boldsymbol{S}}', \underline{\boldsymbol{\varepsilon}})$. Sampling $\boldsymbol{\Theta}$ now involves the modified search distribution $\mathcal{N}(\cdot|\boldsymbol{\mu}, \boldsymbol{S}')$ parameterized by the modified covariance matrix $\boldsymbol{S}'$ and the mean $\boldsymbol{\mu}$ of the true search distribution $\mathcal{N}(\cdot|\boldsymbol{\mu}, \boldsymbol{S})$. After obtaining the dataset, we update the policy with an arbitrary policy search algorithm. Note that we estimate the parameter effectiveness before each sampling step. Therefore, we only use the modified $\boldsymbol{S}'$ for the sampling process and always update the unmodified $\boldsymbol{S}$. Otherwise, the covariance would shrink over time since $\lambda \in (0, 1)$.

2.5 Guided Dimensionality Reduction

To tackle the scalability issue of policy search algorithms, we further exploit our estimation of effective parameters $\bar{\boldsymbol{\varepsilon}}$ to introduce Guided Dimensionality Reduction (GDR). Congruent with our intuition about the varying contribution of parameters to the total return, we now only update the effective ones that promise to increase the total return the most. We again focus on the rotated space and the transformation described in Sec. 2.3.

Running a full covariance policy search update on $\bar{\boldsymbol{S}}_k$ with $\boldsymbol{\Theta}^{rot}$, and $J(\boldsymbol{\Theta})$ gives us the updated covariance matrix $\bar{\boldsymbol{S}}_{k+1} \in \mathbb{R}^{m \times m}$. Note that $\bar{\boldsymbol{S}}_{k+1}$ is no longer guaranteed to be diagonal due to the full covariance update. Then, we compose the new covariance matrix $\boldsymbol{S}_{k+1}$ in the rotated space by substituting the entries of $\bar{\boldsymbol{S}}_k$ with $\bar{\boldsymbol{S}}_{k+1}$, i.e., $\boldsymbol{S}_{k+1} = \text{subst}(\boldsymbol{S}_k, \bar{\boldsymbol{S}}_{k+1}, \bar{\boldsymbol{\varepsilon}})$. We can then project $\boldsymbol{S}_{k+1}$ back into the original space via an inversion of the SVD resulting in $\boldsymbol{\Sigma}_{k+1}$. Analogously, we update $\bar{\boldsymbol{\mu}}_k^{\text{rot}} = \boldsymbol{0}$ corresponding to the effective parameters to obtain $\bar{\boldsymbol{\mu}}_{k+1}^{\text{rot}}$. Substituting the updated values back into $\boldsymbol{\mu}_k^{rot}$ we get $\boldsymbol{\mu}_{k+1}^{rot} = \text{subst}(\boldsymbol{\mu}_k^{rot}, \bar{\boldsymbol{\mu}}_{k+1}^{rot}, \bar{\boldsymbol{\varepsilon}})$. Projecting it back to the original space we yield $\boldsymbol{\mu}_{k+1}$.

Since we choose $m << n$, updating the covariance matrix in the rotated space requires fewer samples than in the original space. Updating only the effective parameters lets us make the most efficient use of these samples without sacrificing performance. Reducing the number of samples makes it viable to use a full covariance matrix, even for high dimensional tasks, assuming an appropriate number of effective parameters is chosen. Furthermore, this approach complements PE perfectly since it maintains a diagonal structure of the ineffective parameters. Assuming similar estimations of the parameter effectiveness at each epoch, the samples would not contain much useful information to update the ineffective parameters as a result of the reduced exploration in exactly those.

2.6 Practical Implementation

The proposed techniques can be applied to a wide variety of BBO algorithms. In Alg. 1 we show how the dimensionality reduction for the full covariance case can be adapted to the CREPS algorithm. In Fig. 1 we provide a visual intuition of the algorithm update and sampling technique. For simplicity, we omit the hyperparameters ϵ and κ, i.e. the KL bound and entropy decrease bound parameters, needed to solve the optimization problems described in Eq. (1) and Eq. (2).

Each update step starts by projecting the full covariance matrix into a space where all dimensions are uncorrelated using SVD. We also rotate the parameter samples $\boldsymbol{\Theta}$ into that space by applying the obtained square rotation matrix $\boldsymbol{U}_k$. We use the rotated samples $\boldsymbol{\Theta}^{\boldsymbol{rot}}$ to compute the $\boldsymbol{C}_{MI}/\boldsymbol{C}_{PCC}$ of each parameter w.r.t. $\boldsymbol{J}$, and determine the set of effective parameters $\bar{\boldsymbol{\varepsilon}}$ and ineffective parameters $\underline{\boldsymbol{\varepsilon}}$. To obtain the optimal value of the Lagrangian multiplier η^*, we minimize the dual function of REPS and use η^* as temperature parameter to compute the weight vector $\boldsymbol{d}_\theta$ for the CWMLE.

To perform CWMLE on the reduced dimensionality, we first create a new diagonal distribution $\mathcal{N}(\cdot|\bar{\boldsymbol{\mu}}_k^{rot}, \bar{\boldsymbol{S}}_{\boldsymbol{k}})$ based on the effective parameters $\bar{\boldsymbol{\varepsilon}}$. We

Algorithm 1 DR-CREPS

procedure UPDATE($\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \boldsymbol{\Theta}, \boldsymbol{J}$)
  $\boldsymbol{U_k}, \boldsymbol{S_k} \leftarrow \text{SVD}(\boldsymbol{\Sigma}_k)$
  $\boldsymbol{\Theta}^{\text{rot}} \leftarrow$ SAMPLEPROJ($\boldsymbol{U}_k, \boldsymbol{\mu}_k, \boldsymbol{\Theta}$), Eq. (6)
  $\boldsymbol{C} \leftarrow$ CORRELATION($\boldsymbol{\Theta}^{\text{rot}}, \boldsymbol{J}$), Eq. (3) or (4)
  $\bar{\boldsymbol{\varepsilon}}, \underline{\boldsymbol{\varepsilon}} \leftarrow$ IDENTIFYEFFECTIVEPARAMETERS($\boldsymbol{C}$)
  $\eta^* \leftarrow$ MINIMIZEDUALFUNCTION($\boldsymbol{\Theta}^{\text{rot}}, \boldsymbol{J}$)
  $\boldsymbol{d}_\theta \leftarrow \exp\left(\frac{\boldsymbol{J}-\max \boldsymbol{J}}{\eta^*}\right)$
  $\boldsymbol{\mu}_{k+1}, \boldsymbol{\Sigma}_{k+1} \leftarrow$ DRCWMLE($\boldsymbol{U}_k, \boldsymbol{S}_k, d_\theta, \bar{\boldsymbol{\varepsilon}}$)
  return $\boldsymbol{\mu}_{k+1}, \boldsymbol{\Sigma}_{k+1}$
end procedure
procedure DRCWMLE($\boldsymbol{U}_k, \boldsymbol{S}_k, d_\theta, \bar{\boldsymbol{\varepsilon}}$)
  $\bar{\boldsymbol{\mu}}_k^{rot}, \bar{\boldsymbol{S}}_k \leftarrow$ GETEFFECTIVEDIST($\boldsymbol{S}_k, \bar{\boldsymbol{\varepsilon}}$)
  $\bar{\boldsymbol{\mu}}_{k+1}^{rot}, \bar{\boldsymbol{S}}_{k+1} \leftarrow$ CWMLE($\bar{\boldsymbol{\mu}}_k^{rot} \bar{\boldsymbol{S}}_k, d_\theta$)
  $\boldsymbol{\mu}_{k+1}^{rot} \leftarrow$ SUBST($\boldsymbol{\mu}_k^{rot}, \bar{\boldsymbol{\mu}}_{k+1}^{rot}, \bar{\boldsymbol{\varepsilon}}$)
  $\boldsymbol{S}_{k+1} \leftarrow$ SUBST($\boldsymbol{S}_k, \bar{\boldsymbol{S}}_{k+1}, \bar{\boldsymbol{\varepsilon}}$)
  $\boldsymbol{\mu}_{k+1}, \boldsymbol{\Sigma}_{k+1} \leftarrow$ BACKPROJ($\boldsymbol{\mu}_{k+1}^{rot}, \boldsymbol{S}_{k+1}$), Eq. (7)
  return $\boldsymbol{\mu}_{k+1}, \boldsymbol{\Sigma}_{k+1}$
end procedure
procedure SAMPLEPE($\boldsymbol{U}_k$, $\boldsymbol{S}_{k+1}$, $\underline{\boldsymbol{\varepsilon}}$)
  $\underline{\boldsymbol{S}}'_{k+1} \leftarrow \lambda \underline{\boldsymbol{S}}_{k+1}$, Eq. (8)
  $\boldsymbol{S}'_{k+1} \leftarrow$ SUBST($\boldsymbol{S}_{k+1}, \underline{\boldsymbol{S}}'_{k+1}, \underline{\boldsymbol{\varepsilon}}$)
  $\boldsymbol{\Sigma}'_{k+1} \leftarrow \boldsymbol{U}_k \boldsymbol{S}'_{k+1} \boldsymbol{U}_k^T$
  $\boldsymbol{\Theta} \sim \mathcal{N}(\cdot|\boldsymbol{\mu}_{k+1}, \boldsymbol{\Sigma}'_{k+1})$
  $\boldsymbol{J} \leftarrow$ Rollout($\boldsymbol{\Theta}$)
  return $\boldsymbol{\Theta}$, $\boldsymbol{J}$
end procedure

then apply the policy search update using the previously computed weights $\boldsymbol{d}_\theta$. Next, we substitute the updated distribution parameters $(\bar{\boldsymbol{\mu}}_{k+1}^{rot}, \bar{\boldsymbol{S}}_{k+1})$ back to obtain the $\mathcal{N}(\cdot|\boldsymbol{\mu}_{k+1}^{rot}, \boldsymbol{S}_{k+1})$. Finally, we rotate the resulting distribution back to the original space yielding a new search distribution $p_{k+1}(\cdot) = \mathcal{N}(\cdot|\boldsymbol{\mu}_{k+1}, \boldsymbol{\Sigma}_{k+1})$.

We apply PE by multiplying $\underline{\boldsymbol{S}}_{k+1}$ with λ before the substitution into $\boldsymbol{S}_{k+1}$. Since $\bar{\boldsymbol{S}}$ and $\underline{\boldsymbol{S}}$ are uncorrelated in the rotated space, scaling $\underline{\boldsymbol{S}}$ by λ will not affect $\bar{\boldsymbol{S}}$. We rotate the sampling covariance $\boldsymbol{S}'_{k+1}$ back to the original space via $\boldsymbol{\Sigma}'_{k+1} = \boldsymbol{U}_k \boldsymbol{S}'_{k+1} \boldsymbol{U}_k^T$ and obtain the proritized samples from $\mathcal{N}(\cdot|\boldsymbol{\mu}_{k+1}, \boldsymbol{\Sigma}'_{k+1})$. As described in Sec. 2.4, we only use $\boldsymbol{\Sigma}'_{k+1}$ for the sampling process, while we still consider $\boldsymbol{\Sigma}_{k+1}$ as target distribution for the KL and entropy bounds.

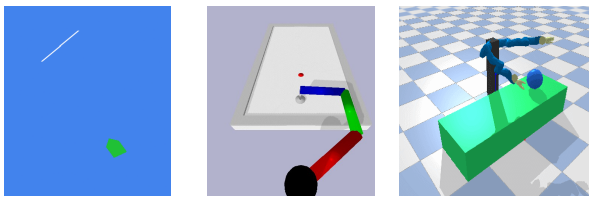
3 EXPERIMENTAL EVALUATION

In this section, we evaluate the proposed method in four different environments. All our implementations are based on the MushroomRL library [D'Eramo et al., 2021]. In all experiments, we use 25 random seeds for evaluation. For each learning curve, we plot the mean and 95% confidence interval. We set a fixed number of episodes for each epoch. The performance is evaluated at the end of each epoch by sampling parameters from the current search distribution. We first test our algorithm in a modified LQR problem where only a part of the controller parameters are effective. Then, we demonstrate several robotic simulations including *ShipSteering* (steering a ship through a gate), *AirHockey* (shooting a goal in a game of air hockey), and *BallStopping* (stopping a ball from rolling off a table) visualized in Fig. 2a, Fig. 2b, and Fig. 2c, respectively. We provide the necessary experimental details to reproduce the experiment in the Supplementary Materials.

3.1 Effect of Prioritized Exploration

To better analyze the behavior of our approach, we introduce a 10-dimensional Linear Quadratic Regulator (LQR) environment that is characterized by three effective and seven ineffective dimensions.We begin our evaluation on the modified LQR and use the diagonal covariance matrix to showcase the performance of PE without GDR.

In Fig. 3, PE shows significantly faster learning when applied to both REPS and CREPS. Especially REPS with PE and $\lambda = 0.1$ profits from the focus on the effective parameters. Reducing the ineffective parameters' covariance to a mere 10% causes the learning curve to jump during the initial epochs. This is a clear sign that the effective parameters were leveraged. However, for REPS this behavior is too greedy, causing premature convergence as also experienced by Pearson-Correlation-Based Relevance Weighted Policy Optimization (PRO) [Ewerton et al., 2019] and Reward-Weighted Regression (RWR) [Deisenroth et al., 2013]. Tuning hyperparameter λ to a higher value allows for some exploration in the other parameters compensating potential mistakes in the parameter identification which would otherwise cause a too greedy behavior. Choosing $\lambda = 0.9$ makes REPS with PE converge to the optimal policy while still outperforming the vanilla version. Note that CREPS with PE does not suffer from this issue due to the additional entropy constraint designed to counteract an excessive greedy behavior, avoiding premature convergence. We provide a more detailed ablation study of hyperparameter λ in the Supplementary Materials.



(a) ShipSteering (b) AirHockey (c) BallStopping

Figure 2: Test environments

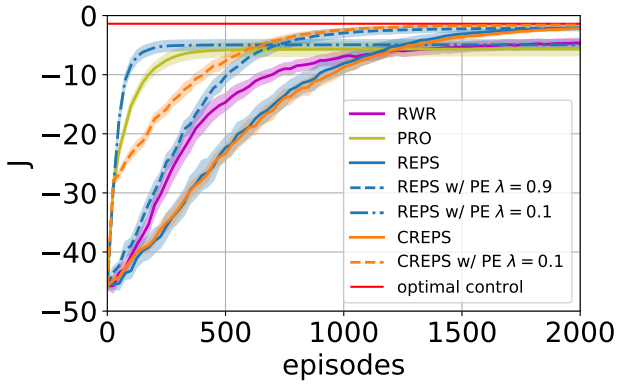


Figure 3: LQR - DiagCov

3.2 Effect of Dimensionality Reduction with Full Covariance Matrix

We demonstrate the learning performance in the LQR environment with full covariance matrix. When using this covariance parametrization, more episodes are required to maintain the positive definiteness during the update. We perform a parameter sweep over the number of episodes per policy update for each algorithm, described in the Supplementary Materials. In Fig. 4, we show the best learning performance of each algorithm. Model-Based Relative Entropy Stochastic Search (MORE) [Abdolmaleki et al., 2015] requires 250 episodes per policy update, CREPS requires 150 episodes, and DR-CREPS requires only 50 episodes. Consequently, DR-CREPS is able to perform more policy updates than MORE and CREPS, resulting in better learning performance. The GDR combined with PE can further exploit the additional updates by focusing the exploration on the effective parameters. The combination of both techniques leads to further improvements in the learning performance.

3.3 Comparison between MI and PCC

We empirically evaluate the effect of MI and PCC as the correlation measures for selecting effective parameters in DR-CREPS on all environments. Both experience similar learning and achieve similar performance, as visualized in Fig. 4 and Fig. 6. To further investigate the performance, we again focus on the modified LQR environment where it is easy to impose a set of effective parameters. In this modified LQR environment, 3 parameters out of 100 are effective parameters. We estimate PCC from Eq. (3) and the MI with scikit-learn's mutual information regressor [Pedregosa et al., 2011]. We compute the precision and recall w.r.t. the parameters and show our results for different numbers of selected parameters $\{10, 30, 50\}$ in Fig. 5. PCC thereby outperforms MI by a short margin on all tested configurations which we accredit to the easier estimation of PCC which does not require density estimation of the distributions over the parameters and the return. Computing the PCC with less samples is therefore expected to achieve a better estimation than MI. Besides reducing the number of samples required for each update, our objective is also to reduce the dimensionality by assuming a small number of effective parameters. This lets us tend towards selecting less parameters, i.e., $20-50\%$ of the total parameters. These propensities explain why PCC and MI have similar learning performances in all experiments with PCC being slightly superior. We provide a more elaborate ablation study of the correlation measure as a comparison to a random selection of the parameters in the Supplementary Materials.

In most environments, the number of ground truth effective parameters is generally not known. It is possible that the number of selected effective parameters is much less than the real number of parameters. Misidentified parameters, however, can prevent overconfident sampling and overly restrictive policy updates, which may lead to premature convergence.

3.4 Learning in Simulated Robotics Environments

The simulated robotic environments are not only more difficult to solve than the LQR but at the same time enable to validate the existence of effective parameters in real-world applications. In all environments, GDR allows updating the policy with fewer episodes than the original CREPS. Together with PE this results in DR-CREPS learning a task-solving policy after only a fraction of the provided episodes.

The *ShipSteering* environment is characterized by a high dimensional policy, i.e., 450 parameters. Learning a full covariance matrix while keeping the positive definiteness using REPS would require an unmanageable number of episodes, and it is therefore not feasible. However, if we combine REPS with GDR, we are able to run the algorithm with only 250 episodes per update. The results in Fig. 6a show that Dimensionality Reduced REPS (DR-REPS) learns a policy on par with CREPS and MORE, while the DR-CREPS algorithm outperforms all other methods.

In *AirHockey*, DR-CREPS learns to score a goal after approx. 2500 episodes indicated by the jump in the total return. Subsequently, it improves its reliability to shoot a goal in the subsequent episodes. MORE, on the other hand, does not show the characteristic jump which is an indicator of the policy not scoring a goal. Here, small differences in the collision between two rounded-shaped objects, i.e., the puck and the end

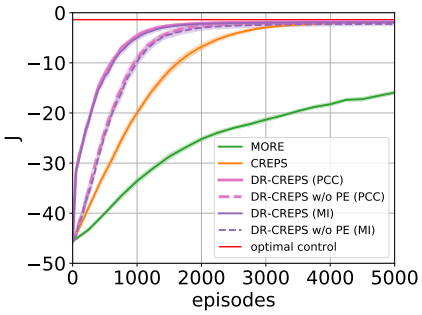


Figure 4: LQR - FullCov

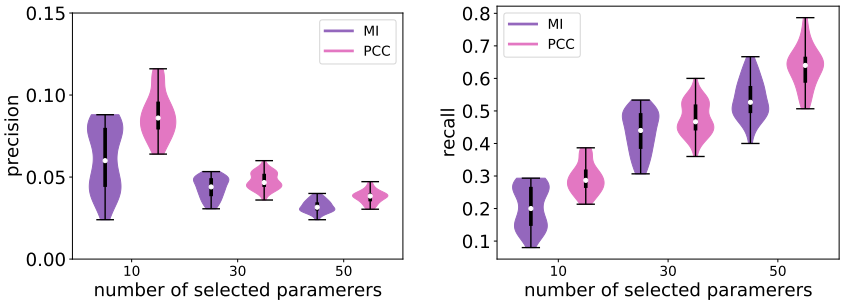
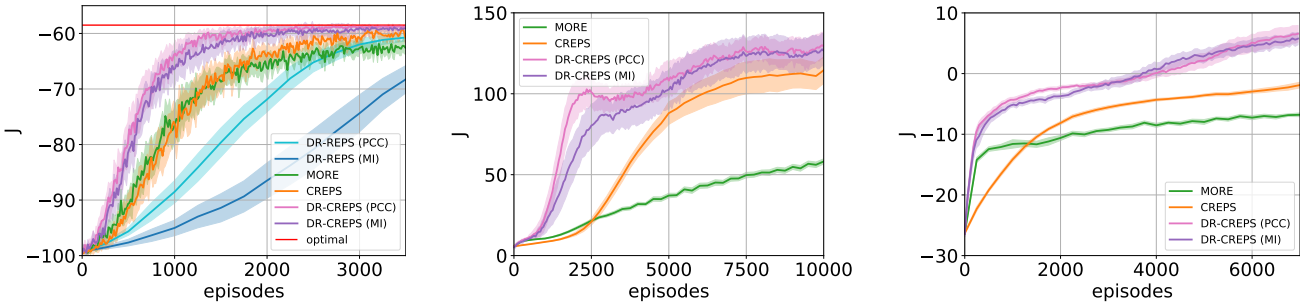


Figure 5: Precision and Recall of Effective Parameter Estimation



(a) ShipSteering (b) AirHockey (c) BallStopping

Figure 6: Experimental results on the simulated environments. DR-CREPS learns faster and converges to better optima than vanilla CREPS and MORE.

effector, result in drastically different trajectories. As a result, the cumulative return has a high variance shown in the learning curve.

DR-CREPS is able to impede the ball after only approx. 500 episodes in the *BallStopping* environment and the final policy even learns to reliably stop it. Observing the learned policies, the advantage of our approach is clear: CREPS moves the arm towards the ball but fails to reliably stops it from rolling off the table while MORE completely fails to move the robotic arm towards the ball or moves the arm too rapidly, propelling the ball back of the table. These behaviors are a result of exploration in the ineffective parameters which cause the arm to act almost randomly and either move it away from the table or let it experience jerky movements. Instead, DR-CREPS reduces the exploration in the ineffective parameters thus, it almost completely prevents those faulty movements leading to a smoother behavior that reliably stops the ball. The GDR then further speeds up the learning by only updating the effective parameters, i.e., the ones required to move the end effector in front of the ball.

4 CONCLUSIONS

In this paper, we propose Guided Dimensionality Reduction, an approach to dimensionality reduction for policy search algorithms in BBO that enables more efficient learning of policies parameterized by full covariance matrices. We utilize the Pearson Correlation Coefficient and the Mutual Information to estimate effective parameters that contribute most to the total return. By updating only those, we reduce the number of samples required for each update allowing our algorithms to boost the learning performance. Furthermore, we present Prioritized Exploration which focuses the exploration on the effective parameters and thereby reduces the exploration space. We empirically demonstrate both methods on DR-CREPS, an adaptation of CREPS, in four different environments including simulated robotics. DR-CREPS outperforms recent approaches to dimensionality reduction in BBO. Future work will focus on bringing the proposed methods to real-world robots and evaluating different Parameter Effectiveness Metrics.

References

[Abdolmaleki et al., 2015] Abdolmaleki, A., Lioutikov, R., Peters, J. R., Lau, N., Pualo Reis, L., and Neumann, G. (2015). Model-based relative entropy stochastic search. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

[Ben Amor et al., 2012] Ben Amor, H., Kroemer, O., Hillenbrand, U., Neumann, G., and Peters, J. (2012). Generalization of human grasping for multi-fingered robot hands. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2043–2050.

[Bitzer et al., 2010] Bitzer, S., Howard, M., and Vijayakumar, S. (2010). Using dimensionality reduction to exploit constraints in reinforcement learning. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3219–3225.

[Bitzer and Vijayakumar, 2009] Bitzer, S. and Vijayakumar, S. (2009). Latent spaces for dynamic movement primitives. In *2009 9th IEEE-RAS International Conference on Humanoid Robots*, pages 574–581.

[Carrara and Ernst, 2019] Carrara, N. and Ernst, J. (2019). On the estimation of mutual information. *Proceedings*, 33(1).

[Choromanski et al., 2018] Choromanski, K., Rowland, M., Sindhwani, V., Turner, R., and Weller, A. (2018). Structured evolution with compact architectures for scalable policy optimization. In *International Conference on Machine Learning*, pages 970–978. PMLR.

[Colomé et al., 2014] Colomé, A., Neumann, G., Peters, J., and Torras, C. (2014). Dimensionality reduction for probabilistic movement primitives. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 794–800.

[Colomé and Torras, 2014] Colomé, A. and Torras, C. (2014). Dimensionality reduction and motion coordination in learning trajectories with dynamic movement primitives. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1414–1420.

[Colomé and Torras, 2018a] Colomé, A. and Torras, C. (2018a). Dimensionality reduction for dynamic movement primitives and application to bimanual manipulation of clothes. *IEEE Transactions on Robotics*, 34(3):602–615.

[Colomé and Torras, 2018b] Colomé, A. and Torras, C. (2018b). Dimensionality reduction in learning gaussian mixture models of movement primitives for contextualized action selection and adaptation. *IEEE Robotics and Automation Letters*, 3(4):3922–3929.

[Deisenroth et al., 2013] Deisenroth, M. P., Neumann, G., Peters, J., et al. (2013). A survey on policy search for robotics. *Foundations and trends in Robotics*, 2(1-2):388–403.

[Delgado-Guerrero et al., 2020] Delgado-Guerrero, J. A., Colomé, A., and Torras, C. (2020). Sample-efficient robot motion learning using gaussian process latent variable models. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 314–320.

[D'Eramo et al., 2021] D'Eramo, C., Tateo, D., Bonarini, A., Restelli, M., and Peters, J. (2021). Mushroomrl: Simplifying reinforcement learning research. *Journal of Machine Learning Research*, 22(131):1–5.

[Ewerton et al., 2019] Ewerton, M., Arenz, O., Maeda, G., Koert, D., Kolev, Z., Takahashi, M., and Peters, J. (2019). Learning trajectory distributions for assisted teleoperation and path planning. *Frontiers in Robotics and AI*, 6:89.

[Furuta et al., 2021] Furuta, H., Matsushima, T., Kozuno, T., Matsuo, Y., Levine, S., Nachum, O., and Gu, S. S. (2021). Policy Information Capacity: Information-Theoretic Measure for Task Complexity in Deep Reinforcement Learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, page 139.

[Jaderberg et al., 2017] Jaderberg, M., Mnih, V., Czarnecki, W. M., Schaul, T., Leibo, J. Z., Silver, D., and Kavukcuoglu, K. (2017). Reinforcement learning with unsupervised auxiliary tasks. In *International Conference on Learning Representations, ICLR 2017*. OpenReview.net.

[Kormushev et al., 2010] Kormushev, P., Calinon, S., and Caldwell, D. G. (2010). Robot motor skill coordination with em-based reinforcement learning. In *2010 IEEE/RSJ international conference on intelligent robots and systems*, pages 3232–3237. IEEE.

[Kraskov et al., 2004] Kraskov, A., Stögbauer, H., and Grassberger, P. (2004). Estimating mutual information. *Phys. Rev. E*, 69:066138.

[Luck et al., 2014] Luck, K. S., Neumann, G., Berger, E., Peters, J., and Amor, H. B. (2014). Latent space policy search for robotics. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1434–1440.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[Peters et al., 2010] Peters, J., Mülling, K., and Altün, Y. (2010). Relative entropy policy search. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, AAAI'10, page 1607–1612. AAAI Press.

[Ploeger et al., 2020] Ploeger, K., Lutter, M., and Peters, J. (2020). High acceleration reinforcement learning for real-world juggling with binary rewards. In *Conference on Robot Learning (CoRL)*.

[Schulman et al., 2015] Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. (2015). Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR.

[Tosatto et al., 2021] Tosatto, S., Chalvatzaki, G., and Peters, J. (2021). Contextual latent-movements off-policy optimization for robotic manipulation skills. In *IEEE International Conference on Robotics and Automation (ICRA)*.