Practical Nonisotropic Monte Carlo Sampling in High Dimensions via Determinantal Point Processes

Krzysztof Choromanski[*] Google Brain Robotics

Aldo Pacchiano[*] UC Berkeley

Jack Parker-Holder[*] University of Oxford

Yunhao Tang[*] Columbia University

Abstract

We propose a new class of practical structured methods for nonisotropic Monte Carlo (MC) sampling, called DPPMC, designed for high-dimensional nonisotropic distributions where samples are correlated to reduce the variance of the estimator via determinantal point processes. We successfully apply DPPMCs to high-dimensional problems involving nonisotropic distributions arising in guided evolution strategy (GES) methods for reinforcement learning (RL), CMA-ES techniques and trust region algorithms for blackbox optimization, improving state-of-the-art in all these settings. In particular, we show that DPPMCs drastically improve exploration profiles of the existing evolution strategy algorithms. We further confirm our results, analyzing random feature map estimators for Gaussian mixture kernels. We provide theoretical justification of our empirical results, showing a connection between DPPMCs and recently introduced structured orthogonal MC methods for isotropic distributions.

1 Introduction

Structured Monte Carlo (MC) sampling has recently received significant attention (Yu et al., 2016; Choromanski et al., 2018b,c, 2017; Rowland et al., 2018; Choromanski et al., 2019c; Rowland et al., 2019) as a universal tool to improve MC methods for applications ranging from dimensionality reduction techniques and random feature map (RFM) kernel approximation Choromanski et al. (2017) Choromanski et al. (2019c) to evolution strategy methods for reinforcement learning (RL) (Rowland et al., 2018, 2019) and estimating sliced Wasserstein distances between high-dimensional probabilistic distributions (Rowland et al., 2019). Structured MC methods rely on choosing samples from joint distributions where different samples are correlated in a particular way to reduce the variance of the estimator. They are also related to the class of $\mathrm{Quasi\ Monte\ Carlo}$ (QMC) methods that aim to improve concentration properties of MC estimators by using low discrepancy sequences of samples to reduce integration error (Yang et al., 2014; Kritzer et al., 2014).

However, the key limitation of the above techniques is that they can only be applied to isotropic distributions, since they rely on samples' orthogonalization. For this class of methods the unbiasedness or asymptotic near-unbiasedness (for large enough dimensionality d) of the resulted orthogonal estimator follows directly from the isotropicity of the corresponding multivariate distribution.

We propose a new class of structured methods for MC sampling, called DPPMC, designed for high-dimensional non-isotropic distributions where samples are correlated to reduce the variance of the estimator via learned or non-adaptive determinantal point processes (DPPs, Kulesza and Taskar (2012); Gartrell et al. (2017)). DPPMCs are designed to work with highly non-isotropic distributions, yet they inherit accuracy gains coming from structured estimators for the isotropic ones. As opposed to other sampling mechanisms applying DPPs (see: Section 2 for more details), we propose a very practical and general framework that can be used in a wide spectrum of scenarios ranging from kernel estimation to reinforcement learning, and is characterized by fast sampling.

We successfully applied DPPMCs to problems involving high-dimensional nonisotropic distributions naturally arising in guided evolution strategy (GES) methods for RL (Maheswaranathan et al., 2019; Choromanski et al., 2019a), CMA-ES techniques and trust region methods for blackbox optimization, improving state-of-the-art in all of these settings. In particular, we show that DPPMCs drastically improve exploration profiles of the existing evolution strategy algorithms. We further confirm our re-

Proceedings of the 23rdInternational Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108. Copyright 2020 by the author(s). [*] Equal contribution.

sults analyzing RFM-estimators for Gaussian mixture kernels (Wilson and Adams, 2013; Samo and Roberts, 2015), and presenting detailed comparison with state-of-the-art density quantization methods. We use MC sampling as a preprocessing step from which a DPP downsamples to construct a final set of samples. Furthermore, we provide theoretical justification of our empirical results, showing a connection between DPPMCs and structured orthogonal MC methods for isotropic distributions.

This paper is organized as follows:**(1)** in Section 2 we discuss related work, **(2)** In Section 3 we introduce Monte Carlo methods and Determinantal Point Processes, **(3)** In Section 4 we introduce our DPPMC algorithm, **(4)** In Section 5 we present theoretical guarantees for the class of DPPMC estimators, **(5)** In Section 6 we present all experimental results, in particular applications to a wide range of reinforcement learning tasks.

Additional experimental details and proofs are given in the Appendix.

2 Related Work

Determinantal Point Processes (DPPs) are becoming increasingly popular in machine learning, as a means to generate diverse subsets of data equipped with elegant mathematical properties and flexible enough to define diversity via general kernel mechanisms. The recent surge of interest in DPPs, has seen research on faster sampling (Gillenwater et al., 2019; Rezaei and Gharan, 2019; Derezinski et al., 2019; Mariet et al., 2019; Li et al., 2016b; Wachinger and Golland, 2015) (e.g. by approximating samples from a DPP with generative deep models) and novel applications. Some works have sought to move away from nonsymmetric DPPs, allowing both attraction and repulsion (Brunel, 2018; Gartrell et al., 2019). Applications of DPPs range from kernel quadrature (Belhadji et al., 2019) to compactifying neural network architectures (the so-called *diversity networks* Mariet and Sra (2016)), dealing with mode collapse in generative models (Elfeki et al., 2019) and improving recommender systems (Gillenwater et al. (2018)).

In parallel, effort has been made to understand whether DPPs can be applied in Monte Carlo integration. A striking result of Bardenet and Hardy (2016) shows that mixing quadratures with repulsive sampling provided by DPPs provably improves convergence rates of MC estimators. Despite providing theoretical foundations for using DPPs in MCs, this result uses expensive machinery of orthogonal multivariate polynomials by relying effectively on *continuous* determinantal point processes. Sampling from them is substantially slower than from "regular" discrete DPPs and in practice cannot be applied to high-dimensional data. The authors of Gautier et al. (2019) build on the results of Bardenet and Hardy (2016) and propose two ways of using DPPs, providing an intriguing connection with the class of Ermakov-Zolotukhin MC estimators. They also manage to speed up the original algorithm by bypassing unnecessary evaluations of the univariate orthogonal Jacobi polynomials. Yet, the resulting algorithm is still slow (all empirical evidence from Gautier et al. (2019) is for data taken from $\mathbb{R}^2$). Sampling from continuous DPPs can be conducted with Markov Chain Monte Carlo (MCMC) techniques (see: Rezaei and Gharan (2019)), but apart from specific classes of kernels those methods are only of theoretical interest due to their time complecity.

To the best of our knowledge, the DPPMC algorithm proposed in this paper is one of the first approaches for using Determinantal Point Processes to conduct *nonisotropic* Monte Carlo sampling with provable theoretical guarantees (see: Section 5) and that can be applied in practice to improve state-of-the-art algorithms operating on high-dimensional data (see: Section 6). In particular, we believe we are the first to show that DPPs can be used to drastically improve Evolution Strategies (ES) algorithms for learning reinforcement learning policies. That exposes this class of techniques to another vibrant field where machine learning can be successully applied: robotics.

Our algorithm is much faster than its related counterparts such as Gautier et al. (2019); Rezaei and Gharan (2019), applied in MC sampling since it does not rely on continuous DPPs and MCMC methods, but on discrete DPPs. It is also easy to implement (see: algorithmic box in Section 4). Efficient (approximate) discrete DPP sampling has been a subject of voluminous literature and several methods have been proposed (see: Gillenwater et al. (2019); Kang (2013); Li et al. (2016a)). As opposed to other sampling mechanisms using DPPs (Li et al., 2016b; Wachinger and Golland, 2015), we propose a general DPP-MC architecture that can be applied in a wide range of downstream scenarios from kernel estimation to reinforcement learning. Presenting downstream accuracy improvements for new important applications of DPPs with respect to state-of-the-art is what distinguishes this work from other recent results focusing more on theoretical aspects of improved DPP samplers, where different DPP methods are benchmarked in terms of their relative speed (Derezinski et al. (2019); Blaszczyszyn and Keeler (2018)).

Our main contributions are though as follows:

- We propose simple to implement algorithm using DPPs for high-dimensional Monte Carlo estimation.
- We apply this algorithm for kernel estimation, reinforcement learning and blackbox optimization, in each setting improving over state-of-the-art.

3 Towards DPPMCs: MC Methods and Determinantal Point Processes

3.1 Unstructured and Structured MC Sampling

Consider a function $F : \mathbb{R}^d \rightarrow \mathbb{R}^m$ defined as follows:

$$F(\theta) = \mathbb{E}_{\mathbf{v} \sim \mathcal{P}}[h_\theta(\mathbf{v})], \tag{1}$$

where: $\mathcal{P} \in \mathcal{P}(\mathbb{R}^d)$ is a distribution from a family of d-dimensional (not necessarily isotropic) distributions and $h_\theta : \mathbb{R}^d \rightarrow \mathbb{R}^m$ is some function. Several important machine learning quantities can be expressed as in Equation 1. For instance, many classes of kernel functions $K : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ admit representation given by Equation 1. The celebrated Bochner's theorem (Rahimi and Recht, 2007) states for every shift-invariant kernel $K : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$:

$$K(\mathbf{x}, \mathbf{y}) = \int_{\mathbf{R}^d} p(\omega) e^{i\omega^T(\mathbf{x}-\mathbf{y})} d\omega, \tag{2}$$

for some distribution $\mathcal{P} \in \mathcal{P}(\mathbb{R}^d)$ with density function p (sometimes called *spectral density*) which is a Fourier Transform of $k : \mathbb{R}^d \rightarrow \mathbb{R}$ defined as $k(\tau) = K(\tau, 0)$. According to Equation 2, values of the stationary kernel K can be written as: $K(\mathbf{x}, \mathbf{y}) = \mathbb{E}_{\mathbf{v} \sim \mathcal{P}}[\cos(\mathbf{v}^\top(\mathbf{x} - \mathbf{y}))]$, for some distribution $\mathcal{P} \in \mathcal{P}(\mathbb{R}^d)$. If furthermore a stationary kernel K is a radial basis function (RBF) kernel, i.e. there exists $g : \mathbb{R} \rightarrow \mathbb{R}$ such that $K(\mathbf{x}, \mathbf{y}) = g(\|\mathbf{x} - \mathbf{y}\|_2)$, then the above distribution is isotropic. RBF kernels include in particular the classes of Gaussian, Matérn and Laplace kernels. Other prominent classes of kernels such as angular kernels or more general *Pointwise Nonlinear Gaussian* kernels (Choromanski et al., 2017) can be also expressed via Equation 1.

Finally, in evolution strategies (ES), a blackbox optimization method frequently applied to learn policies for reinforcement learning and robotics (Salimans et al., 2017; Choromanski et al., 2018c; Rowland et al., 2018; Choromanski et al., 2019b), gradients of Gaussian σ-smoothings of blackbox functions $f : \mathbb{R}^d \rightarrow \mathbb{R}$ (*ES gradients*) are defined as:

$$\nabla_\sigma f(\theta) = \mathbb{E}_{\mathbf{g} \sim \mathcal{N}(0, \mathbf{I}_d)}[\frac{1}{\sigma} f(\theta + \sigma \mathbf{g})\mathbf{g}]. \tag{3}$$

An unbiased baseline MC estimator of $F(\theta)$ from Equation 1 relies on independent sampling from distribution $\mathcal{P}$ and is of the form:

$$\widehat{F}^{\mathrm{iid}}_m = \frac{1}{m} \sum_{i=1}^{m} h_\theta(\mathbf{v}_i), \tag{4}$$

where $\mathbf{v}_i \overset{\mathrm{iid}}{\sim} \mathcal{P}$ and m stands for the number of samples used. In the context of dot-product kernel approximation that estimator leads to the so-called *Johnson-Lindenstrauss Transforms* (Ailon and Liberty, 2011; Dasgupta et al., 2010) and for nonlinear kernel approximation to the celebrated class of random feature map methods (see: Rahimi and Recht (2007)). In blackbox optimization domains it is a core part of many state-of-the-art ES methods (Salimans et al., 2017; Mania et al., 2018; Choromanski et al., 2019b).

In all the above applications distributions $\mathcal{P}$ from which samples were taken are isotropic. For such $\mathcal{P}$, we can further enforce different samples to be exactly orthogonal, while preserving their marginal distributions. This leads to the class of the so-called *orthogonal estimators* $\widehat{F}^{\mathrm{ort}}_m$ (Yu et al., 2016), often characterized by lower variance than their unstructured counterparts (Choromanski et al., 2018b, 2017) followed by downstream gains (in ES optimization (Choromanski et al., 2018c), Wassterstein GAN and autoencoder algorithms (Rowland et al., 2019) or even complicated hybrid predictive state recurrent neural network architectures as in (Choromanski et al., 2018a).

3.2 The Landscape of Nonisotropic Distributions

Two fundamental limitations of the class of estimators $\widehat{F}^{\mathrm{ort}}_m$ is that they need the underlying distributions to be isotropic for their (near)unbiasedness and they require the number of samples to satisfy $m \leq d$. Unfortunately, in practice the number of MC samples m required even for a relatively modest task of spherical Gaussian kernel approximation with precision ϵ with any constant probability is of the order $\Omega(\frac{d}{\epsilon^2}\log(\frac{d}{\epsilon}))$ (see: Rahimi and Recht (2007)). That problem can be addressed by stacking independent orthogonal blocks of samples. However the former problem cannot be solved since the geometry of orthogonal structured transforms is intrinsically intertwined with the isotropicity of $\mathcal{P}$.

Nonisotropic distributions arise in many important applications of machine learning. Several classes of non-RBF kernels are used as a more expressive tool to apply Gaussian processes (GPs) for learning hidden representation in data (Wilson and Adams, 2013). The effectiveness of GPs depends on the quality of the interpolation mechanism applying given kernel function. As noticed in Remes et al. (2017), RBF kernels lead to neighborhood-dominated interpolation that is unable of modelling different parts of the input space in several domains such as: geostatistics, bioinformations, signal processing.

A much more expressive family of non-monotonic (yet still stationary) kernels can be obtained by modelling corresponding spectral density (leading straightforwardly to MC estimators) with the use of Gaussian mixture distributions $\mathcal{P}$ that are no longer isotropic.

To be more specific, take the family of *Gaussian mixture kernels* defined as:

$$K(\mathbf{x},\mathbf{y}) = \sum_{q=1}^{Q} w^q \prod_{i=1}^{d} \exp(-2\pi^2\tau_i^2 v_i^q)\cos(2\pi\tau_i\mu_i^q), \tag{5}$$

where: $\mathbf{x},\mathbf{y} \in \mathbb{R}^d$, $\tau = \mathbf{x}-\mathbf{y}$, Q is the number of Gaussian mixture components, weights w^q define their relative contributions, and finally μ^q and $\mathrm{Cov}^q = \mathrm{diag}(v_1^q,...,v_d^q)$ stand for the mean and covariance matrix of the q^{th} component. The spectral distribution for that class of kernels $\mathcal{P} = \mathcal{N}(\{w^1,\mu^1,\mathrm{Cov}^1\},...,\{w^Q,\mu^Q,\mathrm{Cov}^Q\})$ is a mixture Gaussian distributions with relative weights $\{w^1,...,w^Q\}$, means $\{\mu^1,...,\mu^Q\}$ and covariance matrices $\{\mathrm{Cov}^1,...,\mathrm{Cov}^Q\}$ of different mixture components. Thus the values of these kernels can be expressed as: $K(\mathbf{x},\mathbf{y}) = \mathbb{E}_{\mathbf{v}\sim\mathcal{P}}\cos(\mathbf{v}^\top(\mathbf{x}-\mathbf{y}))$ for the nonisotropic $\mathcal{P}$ defined above.

Since mixtures of Gaussians are dense in the set of distribution functions (in a weak topology sense), by applying Bochner's theorem, we can conclude that Gaussian mixture kernels are dense in the space of all stationary kernels. The generalizations of Gaussian mixture kernels were also proved to be dense in the space of all non-stationary kernels (Samo and Roberts, 2015).

Nonisotropic distributions also play a very important role in blackbox optimization, for instance in the CMA-ES algorithm (Akimoto et al., 2012; Akimoto and Hansen, 2018) to create the populations of samples of parameters to be evaluated in each epoch of the algorithm. Finally, learned nonisotropic distributions are applied on a regular basis in guided ES algorithms for policy optimization (Maheswaranathan et al., 2019; Choromanski et al., 2019a) that estimate gradients of Gaussian smoothings $\nabla_\sigma f(\theta)$ of the RL function f by sampling from nonisotropic distributions.

3.3 Determinantal Point Processes

Consider a finite set of datapoints $\mathcal{X} = \{\mathbf{x}^1,...,\mathbf{x}^N\}$, where $\mathbf{x}^i \in \mathbb{R}^d$. A *determinantal point process* is a distribution $\mathcal{P}$ over the subsets of of $\mathcal{X}$ such that for some real, symmetric matrix $\mathbf{K}$ indexed by the elements of $\mathcal{X}$ the following holds for every $A \subseteq \mathcal{X}$:

$$\mathbb{P}(A \subseteq \mathcal{S}) = \det(\mathbf{K}_A), \tag{6}$$

where $\mathcal{S}$ is sampled from $\mathcal{P}$ and $\mathbf{K}_A$ stands for the submatrix of $\mathbf{K}$ obtained by taking rows and columns indexed by the elements of A. Note that $\mathbf{K}$ is positive semidefinite since all principal minors $\det(\mathbf{K}_A)$ are nonnegative. Determinantal point processes (DPPs) satisfy several so-called *negative dependence property* conditions, such as: $\mathbb{P}[\mathbf{x}^i \in \mathcal{S}|\mathbf{x}^j \in \mathcal{S}] < \mathbb{P}[\mathbf{x}^i \in \mathcal{S}]$ for $i \neq j$, which can be directly derived from their algebraical definition. This makes them an interesting mechanism in applications where the goal is to subsample a diverse set of samples from a given set. To see it even more clearly, we can consider a restricted class of DPPs, the so-called *L-ensembles* (Borodin and Rains, 2005), where the probability that a particular subset S is chosen satisfies:

$$\mathbb{P}[\mathcal{S} = S] = \frac{\det \mathbf{L}_S}{\det(\mathbf{L}+\mathbf{I}_N)} \tag{7}$$

for some matrix $\mathbf{L}$ that as before, has to be positive semidefinite. If we interpret $\mathbf{L}$ as a kernel matrix $\mathbf{L} = [\langle\phi(\mathbf{x}^i),\phi(\mathbf{x}^j)\rangle]_{i,j=1,...,N}$, where ϕ is a corresponding feature map and $\langle\rangle$ stands for the dot-product form in the corresponding Hilbert space, then we see that under the DPP sampling process the sets of near-orthogonal samples in the Hilbert space are favorable over nearly-collinear ones. For instance, if $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^m$ for some $m < \infty$ (as it is the case for example for random feature map representations from Rahimi and Recht (2007)) then probabilities $\mathbb{P}[\mathcal{S} = S]$ are proportional to squared volumes of the parallelepipeds defined by feature vectors $\phi(x^s)$ for $s \in S$. Thus samples that are similar according to a given kernel are less likely to appear together in the subsampled set than those that correspond to the orthogonal elements in the corresponding Hilbert space (see Subsection 5.1).

The DPPs described above construct subsampled sets of different sizes, but if a fixed-size subset is needed a variant of the DPP called a k-DPP can be used (see: Kulesza and Taskar (2011)).

4 DPPMC Algorithm

We propose to estimate the expression from Equation 1 by the following procedure. We first choose the number of samples m that we will average over (as in a standard baseline MC method). We then conduct oversampling by sampling independently at random $m\rho$ samples from $\mathcal{P}$ for some fixed multiplier $\rho > 1$ (which is the hyperparameter of the algorithm) to obtain set S_{MC}. Optionally, we renormalize datapoints of S_{MC} so that they are all of equal lengths. We then downsample from the S_{MC} using m-DPP and get an m-element set $\mathcal{S}_{\mathrm{DPP}}$ (see also: Algorithm box 1 summarizing the method). Finally, we estimate $F(\theta)$ as:

$$\widehat{F}(\theta)^{\mathrm{DPPMC}} = \frac{1}{m}\sum_{\mathbf{v}\in\mathcal{S}_{\mathrm{DPP}}} h_\theta(\mathbf{v}). \tag{8}$$

In most practical applications it suffices to use a DPP determined by a fixed kernel function (see for instance: Mariet and Sra (2016)) and we show in Section 6.2 this approach is successful for RL tasks. However, for completeness we also present a learning framework. In order to learn the right kernel determining matrix $\mathbf{L}$ for the DPP (see: Subsection 3.3), we model this kernel as $K(\mathbf{x},\mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle$, where function ϕ is the output of the feedforward fully connected neural network.

Algorithm 1 DPPMC sampling

```
Input: Distribution $\mathcal{P}$, number of samples $n$, upsample parameter $\rho$, kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$.
1. Sample $\epsilon_1, \cdots, \epsilon_{\rho n}$ independently from $\mathcal{P}$.
2. Use a DPP defined by $K$ to choose diverse subset of $n$ samples $\epsilon_{i_1}, ..., \epsilon_{i_n}$.
3. Evaluate (renormalized) $\epsilon_{i_1}, \cdots, \epsilon_{i_n}$.
```

There is an extensive literature on learning DPPs via learned mappings ϕ produced by neural networks (see: Gartrell et al. (2017)). However, most approaches focus on a different setting, where the goal is to learn the DPP from the subsets it produces (via negative maximal log-likelihood loss functions). Our neural network training is conducted as follows.

We approximate distribution $\mathcal{P}$ by the Gaussian mixture distribution $\mathcal{P}_{\mathrm{GM}}$. In most interesting practical applications the nonisotropic distributions under consideration are already Gaussian mixtures (thus no approximation is needed), but in principle the method can also be applied to other nonisotropic distributions. Then we fix a training set of datapoints $\mathcal{X}_{\mathrm{train}} \subseteq \mathbb{R}^d$. In practice we use publicly available datasets (see: Subsection 6.1) with dimensionalities matching that of distribution $\mathcal{P}$. One can also consider synthetic datasets. Next we train the neural network to minimize the empirical mean squared error (MSE) of the DDPMC estimator of the Gaussian mixture kernel from Equation 5 corresponding to $\mathcal{P}_{\mathrm{GM}}$ on the pairs of points from the training set $\mathcal{X}_{\mathrm{train}}$ (this is just one of many loss functions that can be effectively used here).

For given datapoints $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, the empirical MSE of the DPPMC approximator $\widehat{K}$ of the Gaussian mixture kernel K is given as: $\widehat{\mathrm{MSE}}(\widehat{K}(\mathbf{x},\mathbf{y})) = \frac{1}{t}\sum_{i=1}^{t}[(\frac{1}{m}\sum_{\mathbf{v}\in S^i_{\mathrm{DPP}}} h_\tau(\mathbf{v}) - K(\mathbf{x},\mathbf{y}))^2]$, where $\tau = \mathbf{x} - \mathbf{y}$, $h_\theta(\mathbf{v}) = \cos(\mathbf{v}^\top \theta)$ and sets S^i_{DPP} are constructed by t independent runs of the above procedure, where t is a fixed hyperparameter determining accuracy of the estimation of $\mathrm{MSE}(\widehat{K}(\mathbf{x},\mathbf{y}))$. The final loss function that we backpropagate through is the average empirical MSE over pairs of points from $\mathcal{X}_{\mathrm{train}}$.

The empirical mean squared error of kernels associated with nonisotropic distributions under consideration was chosen on purpose as an objective function minimized during training. For isotropic distributions the orthogonal structure (see: discussion about $\widehat{F}^{\mathrm{ort}}_m$ in Subsection 3.1) that was first introduced as an effective tool for minimizing mean squared error of associated kernels (via random feature map mechanism) was later rediscovered as superior to baseline methods in other downstream tasks, as we discussed in Subsection 3.1.

5 Theoretical Results

In this section we consider functions $F : \mathbb{R}^d \rightarrow \mathbb{R}^m$ from Equation 1. All proofs of the presented results are given in the Appendix. We start by showing that DPPs can be used to provably reduce the MSE of downsampled estimators. Let $\{\mathbf{v}^1, \cdots, \mathbf{v}^N\} \subseteq \mathbb{R}^d$ be N evaluation points of F[1]. Consider the case where each datapoint $\mathbf{v}^i$ is selected as part of the estimator with probability p_i. More formally, let $\{\epsilon_i\}_{i=1}^N$ be an ensemble of Bernoulli random variable with values in $\{0,1\}$ and marginal probabilities $\{p_i\}_{i=1}^N$. Define the unbiased downsampled estimator as:

$$\hat{F}(\theta)_U = \frac{1}{N}\sum_{i=1}^{N} \frac{\epsilon_i}{p_i} h_\theta(\mathbf{v}^i). \tag{9}$$

Notice that $\mathbb{E}_{\{\epsilon_i\}}\left[\hat{F}(\theta)_U\right] = \frac{1}{N}\sum_{i=1}^N h_\theta(\mathbf{v}^i)$. Let $\{w_i\}$ be a set of importance weights with $w_i > 0$. We show that ensembles of Bernoulli random variables $\{\epsilon_i\}$ sampled from a DPP can yield downsampling estimators with better variance than if these are produced i.i.d. with $\epsilon_i \sim \mathrm{Ber}(p_i)$. Let $\mathbf{K}$ be a marginal kernel matrix defining a DPP with marginal probabilities $\mathbf{K}_{i,i} = p_i$ and such that the ensemble follows the DPP process. We consider the following subsampled ES estimator:

$$\hat{F}(\theta)_U^{\mathrm{DPP}} = \frac{1}{N}\sum_{i=1}^{N} \frac{\epsilon_i}{p_i} h_\theta(\mathbf{v}^i), \tag{10}$$

where $\{\epsilon_i\} \sim \mathrm{DPP}(\mathbf{K})$. Recall that here we have: $\mathbb{E}\left[\epsilon_i\right] = \mathbf{K}_{i,i}$ and $\mathbb{E}\left[\epsilon_i\epsilon_j\right] = \mathbf{K}_{i,i}\mathbf{K}_{j,j} - \mathbf{K}_{i,j}^2$ for $i \neq j$. We define $\hat{F}(\theta)_U^{\mathrm{iid}}$ in the analogous way, where this time samples $\{\epsilon_i\}$ are i.i.d. Bernoulli with parameters p_i. In the theorem below we assume that $N \geq d+2$:

Theorem 1. *If $p_i < 1$ for all i, there exists a Marginal Kernel $\mathbf{K} \in \mathbb{R}^{N \times N}$ such that:*

$$\begin{aligned}\mathbb{E}_{\{\epsilon_i\}\sim\mathrm{DPP}(\mathbf{K})}\left[\hat{F}(\theta)_U^{\mathrm{DPP}}\right] &= \mathbb{E}_{\{\epsilon_i\}\sim\{Ber(p_i)\}}\left[\hat{F}(\theta)_U^{\mathrm{iid}}\right] \\ &= \frac{1}{N}\sum_{i=1}^{N} h_\theta(\mathbf{v}^i)\end{aligned} \tag{11}$$

and furthermore: $\mathrm{Var}(\hat{F}(\theta)_U^{\mathrm{DPP}}) < \mathrm{Var}(\hat{F}(\theta)_U^{\mathrm{iid}})$.

[1] An important special case is when $\mathbf{v}^i \sim \mathcal{P}$ for all i although it is not necessary for some of the results in this section to hold.

Thus DPP-based mechanism provides more accurate estimators. As a consequence of the above theorem, we obtain guarantees for estimators of gradients of Gaussian smoothings. Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$ and let $f_\sigma(\theta) = \mathbb{E}_{\mathbf{g} \sim \mathcal{N}(0, \mathbf{I}_d)}[F(\theta + \sigma\mathbf{g})\mathbf{g}]$ be its Gaussian smoothing. Let $\nabla f_\sigma(\theta)$ denote the ES gradient of f, as defined in equation 3, and call $\hat{\nabla}^{\mathrm{iid}}_U f_\sigma(\theta)$ and $\hat{\nabla}^{\mathrm{DPP}}_U f_\sigma(\theta)$ the corresponding unbiased downsampled iid and DPP versions of the estimator of $\nabla f_\sigma(\theta)$.

Corollary 1. *Let $\mathbf{g}^1, \cdots, \mathbf{g}^N \sim \mathcal{N}(0, \mathbf{I}_d)$ be $N \geq d+2$ iid normally distributed perturbations and let $\{p_i\}_{i=1}^N$ such that $p_i < 1$ for all i be an ensemble of downsampling parameters. For any $\theta \in \mathbb{R}^d$ there is a marginal kernel $\mathbf{K} \in \mathbb{R}^{N \times N}$ such that: $\mathbb{E}\left[\hat{\nabla}^{\mathrm{DPP}}_U f_\sigma(\theta)\right] = \mathbb{E}\left[\hat{\nabla}^{\mathrm{iid}}_U f_\sigma(\theta)\right] = \nabla f_\sigma(\theta)$, where: the first expectation is taken with respect to both $\{\mathbf{v}^i\} \sim \mathcal{N}(0, \mathbf{I}_d)$ and $\{\epsilon_i\} \sim \mathrm{DPP}(\mathbf{K})$ and the second expectation is taken with respect to both $\{\mathbf{v}^i\} \sim \mathcal{N}(0, \mathbf{I}_d)$ and $\{\epsilon_i\} \sim \{Ber(p_i)\}$. The variance satisfies: $\mathrm{Var}_(\hat{\nabla}^{\mathrm{DPP}}_U f_\sigma(\theta)) < \mathrm{Var}(\hat{\nabla}^{\mathrm{iid}}_U f_\sigma(\theta))$, where the variance on the LHS of the inequality is computed with respect to $\{\epsilon_i\} \sim \mathrm{DPP}(\mathbf{K})$ and the variance on the RHS is computed with respect to $\{\epsilon_i\} \sim \{Ber(p_i)\}$.*

This implies that provided we select an appropriate DPP-Kernel matrix $\mathbf{K}$, DPPMC yields an unbiased estimator of the gradient of the Gaussian smoothing $\nabla f_\sigma(\theta)$ of smaller variance than iid estimator. The proof of this theorem can be turned into a procedure to produce such a Kernel $\mathbf{K}$. When the probabilities $p_i = p$ for all i, the importance weighted estimator is equivalent (with high probability) to the downsampled estimators we use in Section 6 that already outperform other methods.

5.1 Connections with Orthogonality

In this section we formalize the intuition that the most likely sets sampled under a Determinantal Point Process correspond to subsets of the dataset with orthogonal features in the kernel space. In Choromanski et al. (2018c) the authors study the benefits of coupling sensing directions used to build ES estimators by enforcing orthogonality between the sampling directions while preserving Gaussian marginals. It can be shown this strategy provably reduces the variance of the resulting gradient estimators. We shed light on this phenomenon through the perspective of DPPs. In what follows assume $\mathcal{X} = \{\mathbf{x}^1, \cdots, \mathbf{x}^N\}$ with $\mathbf{x}^i \in \mathbb{R}^d$ and let $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^D$ be a possibly infinite feature map ϕ defining a kernel.

Theorem 2. *Let $\mathbf{L} = [\langle \phi(\mathbf{x}^i), \phi(\mathbf{x}^j) \rangle]_{i,j} \in \mathbb{R}^{N \times N}$ be an $\mathbf{L}-$ ensemble, where $\|\Phi(\mathbf{x}^i)\|_2 = 1$ for all $i \in [N]$. Let $k \in \mathbb{N}$ with $k \leq N$ and assume there exist k samples $\mathbf{x}^{i_1}, \cdots, \mathbf{x}^{i_k}$ in $\mathcal{X}$ satisfying $\langle \phi(\mathbf{x}^{i_j}), \phi(\mathbf{x}^{i_l}) \rangle = 0$ for all $j, l \in [k]$. If $\mathbb{P}_k$ denotes the DPP measure over subsets of size k of $[N]$ defined by $\mathbf{L}$, the most likely outcomes from $\mathbb{P}_k$ are the size-k pairwise orthogonal subsets of $\mathcal{X}$.*

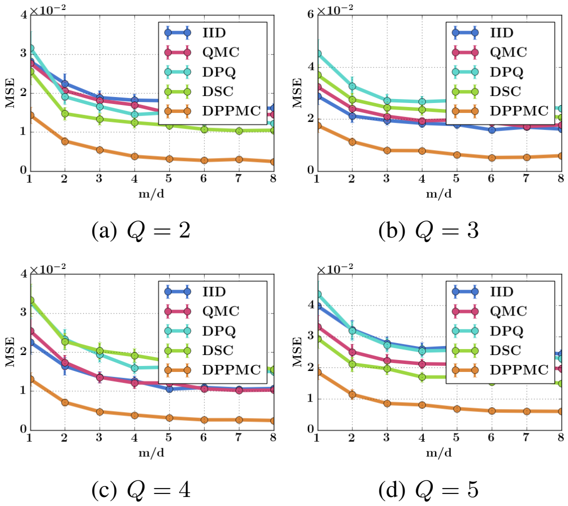
6 Experiments

We aim to address the following questions: **(1)** Do DPPMCs help to achieve better concentration results for MC estimation? **(2)** Do DPPMCs provide benefits for downstream tasks? To address **(1)**, we consider estimating kernels using random features. To address **(2)**, we analyze applications of DPPMCs for high-dimensional blackbox optimization. We present extended ablation studies for parameter ρ in the Appendix (see: Sec. 8.2).

Complexity: We emphasize the conceptual simplicity of our algorithm. Improving state-of-the-art in the RL setting, where we fix an RBF kernel defining the DPP (i.e. learning is not needed) requires adding few lines of code (we include a generic 11-line example of standard DPP python implementation in Section 8.1). Learning a DPP follows the standard supervised framework. Sampling from DPPs requires the eigen-decomposition of matrix $\mathbf{L}$ a priori, however we use fast sub-cubic (k)-DPP approximate sampling mechanisms (Kang, 2013; Li et al., 2016a). For blackbox optimization, time complexity of DPP sampling was negligible in comparison with that for function querying. Thus wall-clock time is accurately measured by the number of timesteps/function evaluations and we show that DPPMC enhancements need substantially fewer of them. For kernel approximation, time complexity of estimating kernel values is exactly the same for the DPPMC and baseline estimator (and reduces to that of matrix-vector multiplication). DPPMC requires DPP sampling, but in that setting it is a one-time cost.

6.1 Kernel Estimation

We compare the accuracy of the baseline MC estimator of values of Gaussian mixture kernels from Equation 5 using independent samples (IID) with those applying Quasi Monte Carlo methods (QMC) Avron et al. (2016), estimators based on state-of-the-art quantization methods: DPQ (Alamgir et al., 2014), DSC (Mirzasoleiman et al., 2015) and our DPPMC mechanism. We applied different QMC estimators and on each plot show the best one. We also compared against other techniques for reducing variance of Monte Carlo estimators such as importance sampling from Elvira et al. (2015) and stratification with antithetic samples and moment-matching methods. Obtained curves are very similar to these for the best QMC variants thus for clarity of the presentation, we do not explicitly present them on the plots. We compare empirical mean squared errors of the above methods. The results are presented on



(a) $Q = 2$

(b) $Q = 3$

(c) $Q = 4$

(d) $Q = 5$

Figure 1: Comparison of different estimators of Gaussian mixture kernels for different number of components: Q on cpu dataset. On the horizontal axis: the ratio of the number of samples used and dimensionality of the datapoints. On the vertical axis: obtained empirical mean squared error.

the enriched cpu dataset. DPP mechanism was trained on the enriched wine dataset. Both datasets were taken from $\mathrm{UCI\ Machine\ LearningRepository}$. Mapping ϕ was encoded by feedforward fully connected neural networks with two hidden layers of size $h = 40$ each and with tanh nonlinearities. We analyzed Gaussian mixture kernels with different number of components Q. Fig. 1 shows that in all settings, DPPMC substantially outperforms all other methods. We did not include orthogonal sampling method, since it did not work for the considered kernels.

6.2 Blackbox Optimization

ES blackbox optimization relies on sampling perturbation directions for function evaluations to optimize sets of parameters (Salimans et al., 2017; Choromanski et al., 2018c). We propose to improve these baseline algorithms by augmenting their sampling subroutines with DPPMCs. We consider the following baseline methods: **(1)** recently proposed guided ES methods, such as Guided Evolution Strategies (Maheswaranathan et al., 2019; Choromanski et al., 2019a), **(2)** Trust-Region based ES methods resusing certain samples for better time complexity (Choromanski et al., 2019b), **(3)** Covariance Matrix Adaptation Evolution Strategy CMA-ES, a state-of-the-art blackbox optimization algorithm (Hansen et al., 2003).

In each setting, the key difference between the baseline algorithm and our proposed method is that the former carries out uniform sampling from a given distribution $\mathcal{P}$, while our method diversifies the set of samples using DPPMC. Using a diverse set of samples leads to more efficient exploration in the parameter space and benefits downstream training, as we show later. We used a fixed Gaussian kernel with tuned variance to determine DPP. We consider two sets of benchmark problems.

Reinforcement Learning: In reinforcement learning (RL), at each time step t an agent observes state $s_t \in \mathcal{S}$, takes action a_t, receives reward $r_t \in \mathbb{R}$ and transitions to the next state $s_{t+1} \in \mathcal{S}$. A policy is a mapping $\pi_\theta : \mathcal{S} \rightarrow \mathcal{A}$ from states to actions that will be conducted in that states and is parameterized by vector θ. The goal is to optimize that mapping to maximize expected cumulative reward $\mathbb{E}[\sum_{t=0}^{T} r_t]$ over given time horizon T. When framing RL as a blackbox optimization problem, the input θ to the blackbox function f is usually a vectorized neural network and the output is a noisy estimate of the cumulative reward, obtained by executing policy π_θ in a particular environment. We consider environments: $\mathrm{Swimmer\text{-}v2}$, $\mathrm{HalfCheetah\text{-}v2}$, $\mathrm{Walker2d\text{-}v2}$ and $\mathrm{Reacher}$ from the $\mathrm{OpenAI\ Gym}$ library and trained policies encoded by fully connected feedforward neural networks.

Nevergrad Functions: Blackbox functions from the recently open-sourced $\mathrm{Nevergrad}$ library (Teytaud and Rapin, 2018), using the well-known open-source implementation of $\mathrm{CMA\text{-}ES}$ (from $\mathrm{https://github.com/CMA\text{-}ES/pycma}$). We tested functions: Cigar, Sphere, $\mathrm{Rosenbrock}$ and $\mathrm{Rastragin}$.

We are ready to describe the considered ES algorithms.

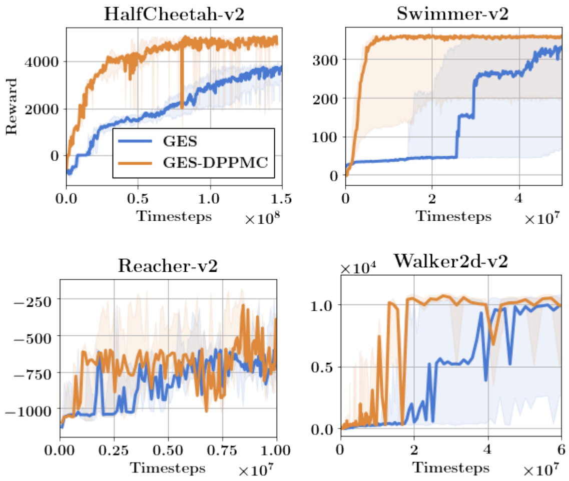


Figure 2: Standard Guided ES versus their DPPMC enhancements on $\mathrm{OpenAI\ Gym}$ tasks. Presented are median-curves from $k = 10$ seeds and with inter-quartile ranges shadowed.

Guided ES: In each iteration, Guided ES methods sample m perturbations from the non-isotropic Gaussian distribution $\mathcal{P}$ with an adaptive covariance matrix computed from the empirical covariance matrix of gradients obtained via a biased oracle or previous estimation, as in recently proposed approaches based on ES-active subspaces (Maheswaranathan et al., 2019; Choromanski et al., 2019a). Such an adaptive non-isotropic sensing often leads to more sample-efficient gradient estimator by exploring subspaces where the true gradients are most likely to be. In the DPPMC enhancement of those techniques, we first sample $l = \rho m$ vectors from $\mathcal{P}$ for $\rho = 10$, and then down-sample to get a subset of m vectors via DPPs.

In Fig.2, we compare baseline Guided ES with its enhanced DPPMC version. The vertical axis shows the expected cumulative reward during training and the horizontal axis - the number of time steps. Each plot shows the average performance with shaded area indicating interquartiles across $r = 10$ random seeds. DPPMC leads to substantially better training curves. To achieve reward ≈ 2000 in HalfCheetah-v2, the baseline algorithm requires $\approx 10^8$ steps while DPPMC only 10^7.

Trust Region ES: Trust Region ES methods, as those recently proposed in Choromanski et al. (2019b), rely on reusing δm perturbations from previous epochs for some $0 < \delta < 1$ and applying regression techniques to estimate gradients of blackbox functions. Those methods do not require perturbations to be independent. DPPMCs can be applied here by sampling $(1 - \frac{\delta}{2})m$ new perturbations (instead of $(1 - \delta)m$) and then downsampling from the set of all $(1 + \frac{\delta}{2})m$ perurbations ($(1 - \frac{\delta}{2})m$ new and δm reused) only m of them. By doing it, we do not reuse all δm samples, but obtain more diverse set of perturbations that improves sampling complexity. We take $\delta = 0.2$.

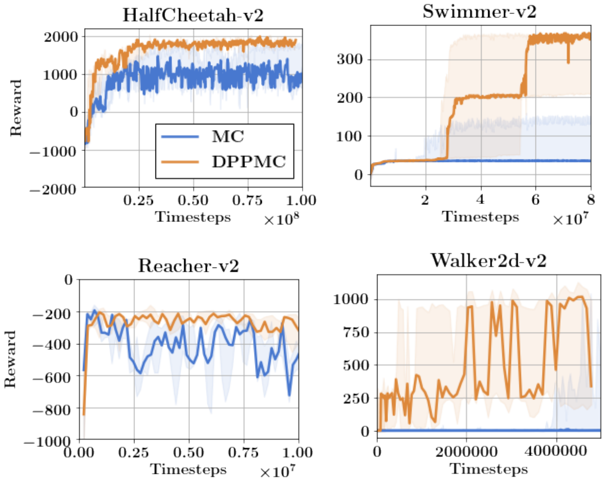


Figure 3: RBO trust region method using MC/ridge gradients versus its DPPMC enhancements on OpenAI Gym tasks. All curves are median-curves from $k = 5$ seeds and with interquartile ranges as shadowed.

As we can see in Fig.3, for most of the cases DPPMC-based Trust Region ES method outperforms algorithm RBO from Choromanski et al. (2019b) that uses standard Trust Region ES mechanism and was already showed to outperform vanilla ES baselines. In particular, for Walker2d-v2 the only method that manages to learn in a given timeframe is based on DPPMC sampling.

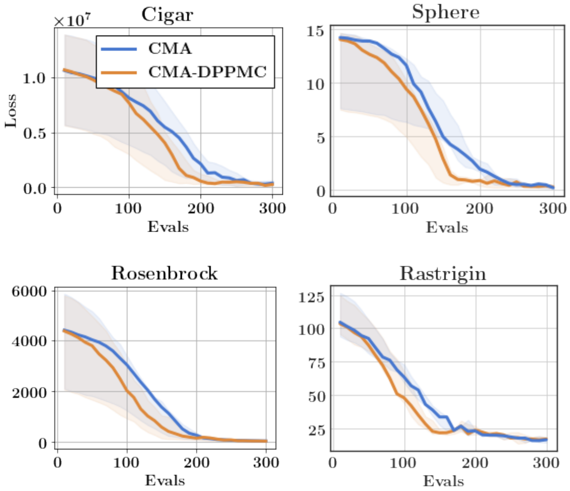


Figure 4: CMA-ES (baseline) versus its DPPMC version for Nevergrad functions. Presented are median-curves from $k = 5$ seeds and with inter-quartile ranges as shadowed.

CMA-ES: In each iteration, CMA-ES samples a set of m perturbation vectors from a non-isotropic Gaussian distribution for function evaluations. Unlike for the above Guided ES methods, the covariance matrix is adapted by running weighted regression over sampled perturbations, where the weights are the function evaluations for different perturbations. Such an adaptive mechanism allows also for efficient exploration in the parameter space, and has performed robustly even for high-dimensional tasks (Hansen et al., 2003; Duan et al., 2016). To construct the candidate pool for CMA-ES, we first sample $l = \rho m$ non-isotropic Gaussian vectors for $\rho = 10$, and then downsample m elements via DPPs.

We compare CMA-ES baseline with its DPPMC enhancement in Fig. 4. The horizontal axis shows the cumulative number of function evaluations we make as the optimization progresses, while the vertical axis shows the expected loss. DPPMC achieves consistent gains across all presented Nevergrad benchmarks. We remark that since the open source implementation of pycma is highly optimized, obtaining even marginal improvements across multiple benchmarks is not trivial.

7 Conclusion

We presented a new sampling mechanism DPPMC based on Determinantal Point Processes to improve standard Monte Carlo methods for nonisotropic distributions. We furthermore showed the effectiveness of our approach on several downstream tasks (guided ES search, CMA-ES and trust-region methods for blackbox optimization) and provided theoretical guarantees.

References

Ailon, N. and Liberty, E. (2011). An almost optimal unrestricted fast Johnson-Lindenstrauss transform. In *SODA*.

Akimoto, Y. and Hansen, N. (2018). CMA-ES and advanced adaptation mechanisms. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO 2018, Kyoto, Japan, July 15-19, 2018*, pages 720–744.

Akimoto, Y., Nagata, Y., Ono, I., and Kobayashi, S. (2012). Theoretical foundation for CMA-ES from information geometry perspective. *Algorithmica*, 64(4):698–716.

Alamgir, M., Lugosi, G., and Luxburg, U. (2014). Density-preserving quantization with application to graph downsampling. In Balcan, M. F., Feldman, V., and Szepesvri, C., editors, *Proceedings of The 27th Conference on Learning Theory*, volume 35 of *Proceedings of Machine Learning Research*, pages 543–559, Barcelona, Spain. PMLR.

Avron, H., Sindhwani, V., Yang, J., and Mahoney, M. W. (2016). Quasi-Monte Carlo feature maps for shift-invariant kernels. *The Journal of Machine Learning Research*, 17(1):4096–4133.

Bardenet, R. and Hardy, A. (2016). Monte carlo with determinantal point processes.

Belhadji, A., Bardenet, R., and Chainais, P. (2019). Kernel quadrature with dpps. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Blaszczyszyn, B. and Keeler, H. P. (2018). Determinantal thinning of point processes with network learning applications. *CoRR*, abs/1810.08672.

Borodin, A. and Rains, E. (2005). Eynardmehta theorem, schur process, and their pfaffian analogs. In *Journal of Statistical Physics*, page 291317.

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

Brunel, V.-E. (2018). Learning signed determinantal point processes through the principal minor assignment problem. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31*, pages 7365–7374. Curran Associates, Inc.

Choromanski, K., Downey, C., Boots, B., Holtmann-Rice, D., and Kumar, S. (2018a). Initialization matters: Orthogonal predictive state recurrent neural networks. In *ICLR*.

Choromanski, K., Pacchiano, A., Parker-Holder, J., and Tang, Y. (2019a). From complexity to simplicity: Adaptive es-active subspaces for blackbox optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Choromanski, K., Pacchiano, A., Parker-Holder, J., Tang, Y., Jain, D., Yang, Y., Iscen, A., Hsu, J., and Sindhwani, V. (2019b). Provably robust blackbox optimization for reinforcement learning. In *Conference on Robot Learning*.

Choromanski, K., Pacchiano, A., Pennington, J., and Tang, Y. (2019c). Kama-nns: low-dimenaional rotation-based neural networks. In *International Conference on Artificial Intelligence and Statistics, AISTATS*.

Choromanski, K., Rowland, M., Sarlos, T., Sindhwani, V., Turner, R., and Weller, A. (2018b). The geometry of random features. In *AISTATS 2018*.

Choromanski, K., Rowland, M., Sindhwani, V., Turner, R. E., and Weller, A. (2018c). Structured evolution with compact architectures for scalable policy optimization. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pages 969–977.

Choromanski, K. M., Rowland, M., and Weller, A. (2017). The unreasonable effectiveness of structured random orthogonal embeddings. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 218–227.

Dasgupta, A., Kumar, R., and Sarlós, T. (2010). A sparse Johnson-Lindenstrauss transform. In *STOC*.

Derezinski, M., Calandriello, D., and Valko, M. (2019). Exact sampling of determinantal point processes with sublinear time preprocessing. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Duan, Y., Chen, X., Houthooft, R., Schulman, J., and Abbeel, P. (2016). Benchmarking deep reinforcement learning for continuous control. In *International Conference on Machine Learning*, pages 1329–1338.

Elfeki, M., Couprie, C., Riviere, M., and Elhoseiny, M. (2019). GDPP: Learning diverse generations using determinantal point processes. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1774–1783, Long Beach, California, USA. PMLR.

Elvira, V., Martino, L., Luengo, D., and Bugallo, M. F. (2015). Efficient multiple importance sampling estimators. *IEEE Signal Process. Lett.*, 22(10):1757–1761.

Gartrell, M., Brunel, V.-E., Dohmatob, E., and Krichene, S. (2019). Learning nonsymmetric determinantal point processes. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Gartrell, M., Paquet, U., and Koenigstein, N. (2017). Low-rank factorization of determinantal point processes. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 1912–1918.

Gautier, G., Bardenet, R., and Valko, M. (2019). On two ways to use determinantal point processes for monte carlo integration. In *Advances in Neural Information Processing Systems*.

Gillenwater, J., Kulesza, A., Mariet, Z., and Vassilvtiskii, S. (2019). A tree-based method for fast repeated sampling of determinantal point processes. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2260–2268, Long Beach, California, USA. PMLR.

Gillenwater, J. A., Kulesza, A., Vassilvitskii, S., and Mariet, Z. E. (2018). Maximizing induced cardinality under a determinantal point process. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pages 6911–6920.

Hansen, N., Müller, S. D., and Koumoutsakos, P. (2003). Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evol. Comput.*, 11(1):1–18.

Kang, B. (2013). Fast determinantal point process sampling with application to clustering. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 2319–2327.

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

Kritzer, P., Niederreiter, H., Pillichshammer, F., and Winterhof, A., editors (2014). *Uniform Distribution and Quasi-Monte Carlo Methods - Discrepancy, Integration and Applications*, volume 15 of *Radon Series on Computational and Applied Mathematics*. De Gruyter.

Kulesza, A. and Taskar, B. (2011). k-dpps: Fixed-size determinantal point processes. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 1193–1200.

Kulesza, A. and Taskar, B. (2012). Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, 5(2-3):123–286.

Li, C., Jegelka, S., and Sra, S. (2016a). Efficient sampling for k-determinantal point processes. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016, Cadiz, Spain, May 9-11, 2016*, pages 1328–1337.

Li, C., Jegelka, S., and Sra, S. (2016b). Fast DPP sampling for nystrom with application to kernel methods. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 2061–2070.

Maheswaranathan, N., Metz, L., Tucker, G., Choi, D., and Sohl-Dickstein, J. (2019). Guided evolutionary strategies: augmenting random search with surrogate gradients. *ICML*.

Mania, H., Guy, A., and Recht, B. (2018). Simple random search provides a competitive approach to reinforcement learning. *CoRR*, abs/1803.07055.

Mariet, Z., Ovadia, Y., and Snoek, J. (2019). Dppnet: Approximating determinantal point processes with deep networks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Mariet, Z. and Sra, S. (2016). Diversity networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Mirzasoleiman, B., Karbasi, A., Badanidiyuru, A., and Krause, A. (2015). Distributed submodular cover: Succinctly summarizing massive data. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2881–2889.

Moritz, P., Nishihara, R., Wang, S., Tumanov, A., Liaw, R., Liang, E., Elibol, M., Yang, Z., Paul, W., Jordan, M. I., et al. (2018). Ray: A distributed framework for emerging {AI} applications. In *13th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 18)*, pages 561–577.

Rahimi, A. and Recht, B. (2007). Random features for large-scale kernel machines. In *NIPS*.

Remes, S., Heinonen, M., and Kaski, S. (2017). Non-stationary spectral kernels. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 4645–4654.

Rezaei, A. and Gharan, S. O. (2019). A polynomial time MCMC method for sampling from continuous determinantal point processes. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5438–5447, Long Beach, California, USA. PMLR.

Rowland, M., Choromanski, K., Chalus, F., Pacchiano, A., Sarlós, T., Turner, R. E., and Weller, A. (2018). Geometrically coupled monte carlo sampling. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pages 195–205.

Rowland, M., Hron, J., Tang, Y., Choromanski, K., Sarlos, T., and Weller, A. (2019). Orthogonal estimation of wasserstein distances. In *International Conference on Artificial Intelligence and Statistics, AISTATS*.

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

Samo, Y.-L. K. and Roberts, S. J. (2015). Generalized spectral kernels. In *arXiv:1506.02236*.

Teytaud, O. and Rapin, J. (2018). Nevergrad: An open source tool for derivative-free optimization. *https://code.fb.com/ai-research/nevergrad/*.

Van Der Walt, S., Colbert, S. C., and Varoquaux, G. (2011). The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22.

Wachinger, C. and Golland, P. (2015). Sampling from determinantal point processes for scalable manifold learning. In *Inf Process Med Imaging*, page 687698.

Wilson, A. G. and Adams, R. P. (2013). Gaussian process kernels for pattern discovery and extrapolation. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, pages 1067–1075.

Yang, J., Sindhwani, V., Avron, H., and Mahoney, M. W. (2014). Quasi-monte carlo feature maps for shift-invariant kernels. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 485–493.

Yu, F., Suresh, A., Choromanski, K., Holtmann-Rice, D., and Kumar, S. (2016). Orthogonal random features. In *NIPS*, pages 1975–1983.