Structured Monte Carlo Sampling for Nonisotropic Distributions via Determinantal Point Processes

Krzysztof Choromanski[*]
Google Brain Robotics
kchoro@google.com

Aldo Pacchiano[*]
UC Berkeley
pacchiano@berkeley.edu

Jack Parker-Holder[*]
Columbia University
jh3764@columbia.edu

Yunhao Tang[*]
Columbia University
yt2541@columbia.edu

Abstract

We propose a new class of structured methods for 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 problems involving non-isotropic distributions arising in guided evolution strategy (GES) methods for 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 structured orthogonal MC methods for isotropic distributions.

1 Introduction

Structured Monte Carlo (MC) sampling has recently received significant attention [42, 12, 13, 14, 33, 11, 34] as a universal tool to improve MC methods for applications ranging from dimensionality reduction techniques and random feature map (RFM) kernel approximation [14, 11] to evolution strategy methods for reinforcement learning (RL) [33, 34] and estimating sliced Wasserstein distances between high-dimensional probabilistic distributions [34]. 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 *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 [41, 21].

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) [23, 17]. 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 using DPPs [25, 39], we propose a general hybrid DPP-MC architecture that can be applied in a wide range of scenarios from kernel estimation to RL.

$*$ Equal Contribution.

We successfully applied DPPMCs to problems involving high-dimensional nonisotropic distributions naturally arising in guided evolution strategy (GES) methods for RL [26], 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 results analyzing RFM-estimators for Gaussian mixture kernels [40, 36], 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.

To motivate our approach, we mention the striking result from [6] showing that mixing quadratures with repulsive sampling provided by DPPs provably improves convergence rates of MC estimators. However, our algorithm is different - we do not rely on sampling from DPPs associated with multivariate orthogonal polynomials which requires cubic time. To the best of our knowledge, we are also the first to provide an extensive empirical evaluation showing that our approach is not only theoretically sound, but leads to efficient algorithms across a variety of settings.

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

2 Towards DPPMCs: MC Methods and Determinantal Point Processes

2.1 Unstructured and Structured MC Sampling

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

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

where: $\mathcal{D} \in \mathcal{P}(\mathbb{R}^d)$ is a d-dimensional (not necessarily isotropic) distribution 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 [31] states for every shift-invariant kernel $K : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$:

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

for some distribution $\mathcal{D} \in \mathcal{P}(\mathbb{R}^d)$ with density function $p_{\mathcal{D}}$ (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: $K(\mathbf{x}, \mathbf{y}) = \mathbb{E}_{\mathbf{v} \sim \mathcal{D}}[\cos(\mathbf{v}^\top(\mathbf{x} - \mathbf{y}))]$, for some distribution $\mathcal{D} \in \mathcal{P}(\mathbb{R}^d)$. 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 [14] 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 [35, 13, 33, 10], 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 $\mathcal{D}$ and is of the form:

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

where $\mathbf{v}_i \overset{\mathrm{iid}}{\sim} \mathcal{D}$ 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* [1, 15] and for

nonlinear kernel approximation to the celebrated class of random feature map methods (see: [31]). In blackbox optimization domains it is a core part of many state-of-the-art ES methods [35, 27, 10].

In all the above applications distributions $\mathcal{D}$ from which samples were taken are isotropic. For such $\mathcal{D}$, 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* $\widehat{F}_m^{\mathrm{ort}}$ [42], often characterized by lower variance than their unstructured counterparts [12, 14] followed by downstream gains (in ES optimization [13], Wasserstein GAN and autoencoder algorithms [34] or even complicated hybrid predictive state recurrent neural network architectures as in [9]).

2.2 The Landscape of Nonisotropic Distributions

Two fundamental limitations of the class of estimators $\widehat{F}_m^{\mathrm{ort}}$ 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: [31]). 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 $\mathcal{D}$.

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 [40]. The effectiveness of GPs depends on the quality of the interpolation mechanism applying given kernel function. As noticed in [32], 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 $\mathcal{D}$ 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 $\mathcal{D} = \mathcal{N}(\{w^1,\mu^1,\mathrm{Cov}^1\},...,\{w^Q,\mu^Q,\mathrm{Cov}^Q\})$ 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: $K(\mathbf{x},\mathbf{y}) = \mathbb{E}_{\mathbf{v}\sim\mathcal{D}}\cos(\mathbf{v}^\top(\mathbf{x}-\mathbf{y}))$ for the nonisotropic $\mathcal{D}$ 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 [36].

Nonisotropic distributions also play a very important role in blackbox optimization, for instance in the CMA-ES algorithm [3, 2] 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 [26] that estimate gradients of Gaussian smoothings $\nabla_\sigma f(\theta)$ of the RL function f by sampling from nonisotropic distributions.

2.3 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* [7], 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 $\mathbf{L} = [\langle \phi(\mathbf{x}^i), \phi(\mathbf{x}^j) \rangle]_{i,j=1,\ldots,N}$, 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 [31]) 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 also Subsection 4.1).

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: [22]).

3 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 $\mathcal{D}$ 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 $\mathcal{S}_{\mathrm{DPP}}$. 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: [28]) and we show in Section 5.2 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 2.3), 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.

There is an extensive literature on learning DPPs via learned mappings ϕ produced by neural networks (see: [17]). 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 $\mathcal{D}$ by the Gaussian mixture distribution $\mathcal{D}_{\mathrm{GM}}$. 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 5.1) with dimensionalities matching that of distribution $\mathcal{D}$. 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 $\mathcal{D}_{\mathrm{GM}}$ 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 2.1) 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 2.1.

4 Theoretical Results

In this section we consider functions $F : \mathbb{R}^d \to \mathbb{R}^m$ 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:

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

where $\{\epsilon_i\} \sim \mathrm{DPP}(\mathbf{K})$. Recall that here we have: $\mathbb{E}[\epsilon_i] = \mathbf{K}_{i,i}$ and $\mathbb{E}[\epsilon_i \epsilon_j] = \mathbf{K}_{i,i}\mathbf{K}_{j,j} - \mathbf{K}^2_{i,j}$ for $i \neq j$. We define $\hat{F}(\theta)^{\mathrm{iid}}_U$ 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 4.1. *If $p_i < 1$ for all i, there exists a Marginal Kernel $\mathbf{K} \in \mathbb{R}^{N \times N}$ such that:*

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

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

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 $f : \mathbb{R}^d \to \mathbb{R}$ 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 4.1. *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: $\mathbb{E}\left[\hat{\nabla}^{\mathrm{DPP}}_U f_\sigma(\theta)\right] = \mathbb{E}\left[\hat{\nabla}^{\mathrm{iid}}_U f_\sigma(\theta)\right] = \nabla f_\sigma(\theta)$,*

$$\underbrace{\mathbb{E}\left[\hat{\nabla}^{\mathrm{DPP}}_U f_\sigma(\theta)\right]}_{I} = \underbrace{\mathbb{E}\left[\hat{\nabla}^{\mathrm{iid}}_U f_\sigma(\theta)\right]}_{II} = \nabla f_\sigma(\theta),$$

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

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:* $\mathrm{Var}_($

$$\underbrace{\mathrm{Var}_(\hat{\nabla}_U^{\mathrm{DPP}} f_\sigma(\theta))}_{I} < \underbrace{\mathrm{Var}(\hat{\nabla}_U^{\mathrm{iid}} f_\sigma(\theta))}_{II},$$

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 5 that already outperform other methods.

4.1 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 [13] 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 $\phi : \mathbb{R}^d \to \mathbb{R}^D$ be a possibly infinite feature map ϕ defining a kernel.

Theorem 4.2. *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* $\mathcal{X}$*.*

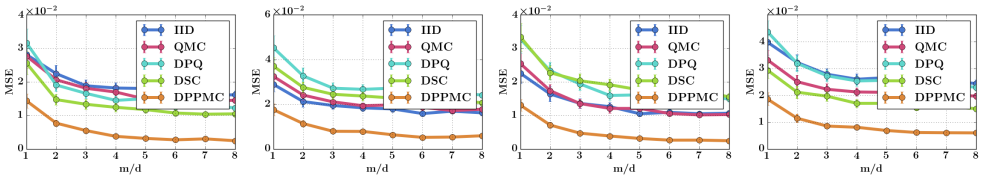
5 Experiments

We aim to address here 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 regarding parameter ρ in the Appendix (see: Section 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 a priori the eigen-decomposition of matrix $\mathbf{L}$, however we use fast sub-cubic (k)-DPP sampling mechanisms [19, 24]. 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.

5.1 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) [5], estimators based on state-of-the-art quantization methods: DPQ [4], DSC [29] and our DPPMC mechanism. We applied different QMC estimators and on each plot show the best one. We compare empirical mean squared errors of the above methods. The results are presented on cpu dataset. DPP mechanism was trained on wine dataset. Mapping ϕ was encoded by standard feedforward fully



(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.

connected neural network architectures 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.

5.2 Blackbox Optimization

ES blackbox optimization algorithms rely on sampling perturbation directions for function evaluations to optimize sets of parameters [35, 13]. 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 [26], **(2)** Trust-Region based ES methods resusing certain samples for better time complexity [10], **(3)** Covariance Matrix Adaptation Evolution Strategy CMA-ES, a state-of-the-art blackbox optimization algorithm [18].

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 $\mathcal{D}$, 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: Swimmer-v2, HalfCheetah-v2, Walker2d-v2 and Reacher from the OpenAI Gym library and trained policies encoded by fully connected feedforward neural networks.

Nevergrad Functions: Blackbox functions from the recently open-sourced Nevergrad library [37], using the well-known open-source implementation of CMA-ES (from https://github.com/CMA-ES/pycma). We tested functions: Cigar, Sphere, Rosenbrock and Rastragin.

We are ready to describe the considered ES algorithms.

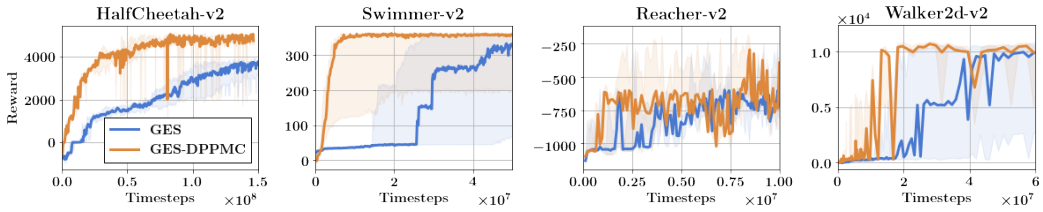


Figure 2: Standard Guided ES versus their DPPMC enhancements on OpenAI Gym tasks. Presented are median-curves from $k = 10$ seeds and with inter-quartile ranges as shadowed regions.

Guided ES: In each iteration, Guided ES methods sample m perturbation vectors from the non-isotropic Gaussian distribution $\mathcal{D}$ with an adaptive covariance matrix computed from the empirical covariance matrix of gradients obtained via a biased oracle [26] or previous estimation, as it is the case in recently proposed approaches based on ES-active subspaces. Such an adaptive non-isotropic sensing often leads to more sample-efficient estimation of the gradient 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 $\mathcal{D}$ for $\rho = 10$, and 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 inter-quartiles across $r = 10$ random seeds. DPPMC leads to substantially better training curves. To achieve reward ≈ 2000 in HalfCheetah-v2, baseline algorithm requires $\approx 10^8$ steps while DPPMC only 10^7.

Trust Region ES: Trust Region ES methods, as those recently proposed in [10], 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 in this setting 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 much more diverse set of perturbations that ultimately improves sampling complexity. We take $\delta = 0.2$.

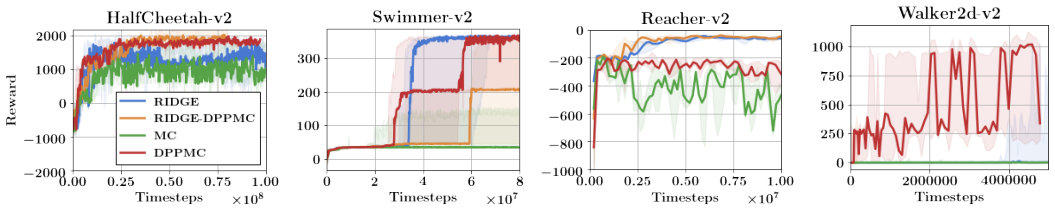


Figure 3: RBO trust region method using MC/ridge gradients versus its DPPMC enhancements on OpenAI Gym tasks. All curves are median-curves from $k = 5$ seeds and with inter-quartile ranges as shadowed regions.

As we can see in Fig.3, for most of the cases DPPMC-based Trust Region ES method outperforms algorithm RBO from [10] 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.

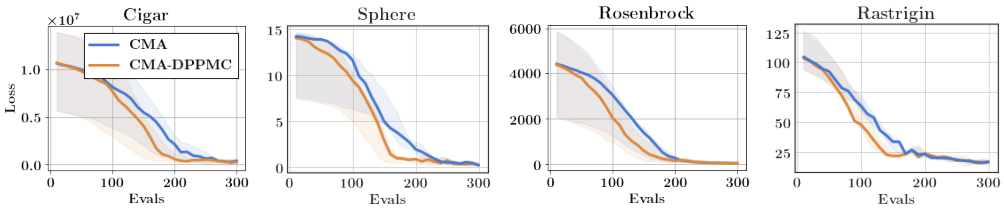


Figure 4: CMA-ES (baseline) versus its DPPMC version for Nevergrad functions. Presented are median-curves from $k = 5$ seeds and with inter-quartile ranges as shadowed regions.

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 [18, 16]. 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. Each plot shows the average performance with shaded

area indicating inter-quartiles across 5 random seeds. 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.

6 Conclusion

We presented new sampling mechanism DPPMC based on determinantal point processes to improve standard MC 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

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

[2] Y. Akimoto and N. Hansen. 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, 2018.

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

[4] M. Alamgir, G. Lugosi, and U. Luxburg. Density-preserving quantization with application to graph downsampling. In M. F. Balcan, V. Feldman, and C. Szepesvri, editors, *Proceedings of The 27th Conference on Learning Theory*, volume 35 of *Proceedings of Machine Learning Research*, pages 543–559, Barcelona, Spain, 13–15 Jun 2014. PMLR.

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

[6] R. Bardenet and A. Hardy. Monte carlo with determinantal point processes. 2016.

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

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

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

[10] K. Choromanski, A. Pacchiano, J. Parker-Holder, J. Hsu, A. Iscen, D. Jain, and V. Sindhwani. When random search is not enough: Sample-efficient and noise-robust blackbox optimization of rl policies. In *https://arxiv.org/pdf/1903.02993.pdf*, 2019.

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

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

[13] K. Choromanski, M. Rowland, V. Sindhwani, R. E. Turner, and A. Weller. 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, 2018.

[14] K. M. Choromanski, M. Rowland, and A. Weller. 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, 2017.

[15] A. Dasgupta, R. Kumar, and T. Sarlós. A sparse Johnson-Lindenstrauss transform. In *STOC*, 2010.

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

[17] M. Gartrell, U. Paquet, and N. Koenigstein. 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, 2017.

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

[19] B. Kang. 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, 2013.

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

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

[22] A. Kulesza and B. Taskar. 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, 2011.

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

[24] C. Li, S. Jegelka, and S. Sra. 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, 2016.

[25] C. Li, S. Jegelka, and S. Sra. 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, 2016.

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

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

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

[29] B. Mirzasoleiman, A. Karbasi, A. Badanidiyuru, and A. Krause. 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, 2015.

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

[31] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *NIPS*, 2007.

[32] S. Remes, M. Heinonen, and S. Kaski. 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, 2017.

[33] M. Rowland, K. Choromanski, F. Chalus, A. Pacchiano, T. Sarlós, R. E. Turner, and A. Weller. 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, 2018.

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

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

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

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

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

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

[40] A. G. Wilson and R. P. Adams. 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, 2013.

[41] J. Yang, V. Sindhwani, H. Avron, and M. W. Mahoney. 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, 2014.

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

7 APPENDIX: Structured Monte Carlo Sampling for Nonisotropic Distributions via Determinantal Point Processes

7.1 Variance Reduction for Evolution Strategies using DPPs

The goal of this section is to show that it is possible to use DPPs to reduce the variance of Evolution Strategies gradient estimators.

7.1.1 One dimensional variance reduction using DPPs

We start by showing an auxiliary sequence of one dimensional lemmas. We consider the problem of computing an estimator of the sum $\bar{a}$ of n real numbers $a_1, \cdots, a_n$. In Lemma 7.1 we first show that using DPPs it is always possible to produce an unbiased estimator of the sum of a sequence of real numbers with less or equal variance than the i.i.d estimator that samples each element a_i of the sequence i.i.d. with probability p_i. We then show in Lemma 7.2 that it is possible to produce a DPP kernel $\mathbf{K}$ such that the corresponding sum estimator has strictly less variance than the i.i.d. one.

We follow the discussion regarding Determinantal Point Process from [23]. Recall that a Determinantal Point Pricess (DPP) $\mathcal{P}$ on a ground set $\mathcal{X}$ with $|\mathcal{X}| = N$ is a probability measure over power set $2^{\mathcal{X}}$. When $\mathcal{S}$ is a random subset drawn according to $\mathcal{P}$, we have, for every $A \subset \mathcal{X}$.

$$\mathcal{P}\left(A \subset \mathcal{S}\right) = \det\left(\mathbf{K}_A\right)$$

for some real symmetric $N \times N$ matrix $\mathbf{K}$ indexed by the elements of $\mathcal{X}$. Here $\mathbf{K}_A = [\mathbf{K}_{i,j}]_{i,j \in A}$ and adopt $\det(\mathbf{K}_\emptyset) = 1$. $\mathbf{K}$ is known as the marginal kernel.

Notice that whenever $A = \{i\}$, $\mathbb{P}(i \in \mathbf{S}) = \mathbf{K}_{i,i}$ and that $\mathbb{P}(i, j \in \mathcal{S}) = \mathbb{P}(i \in \mathcal{S})\mathbb{P}(j \in \mathcal{S}) - \mathbf{K}_{i,j}^2$.

We start by showing a basic variance reduction result regarding DPPs. Let $a_1, \cdots, a_n$ be set of real numbers. Let $\bar{a}$ be their sum. We are interested in analyzing the following two estimators of $\bar{a}$:

1. $\hat{a}_{\text{i.i.d}} = \sum_{i=1}^{n} \frac{a_i \epsilon_i}{p_i}$ where ϵ_i are sampled independent from each other with $\epsilon_i \sim \text{Ber}(p_i)$.
2. $\hat{a}_{\text{DPP}} = \sum_{i \in \mathcal{S}} \frac{a_i \epsilon_i}{p_i}$ where $\mathcal{S}$ is a subset of $[n]$ sampled from a DPP with kernel $\mathbf{K}$ satisfying $\mathbf{K}_{i,i} = p_i$ for all i.

Notice that $\mathbb{E}\left[\hat{a}_{\text{i.i.d}}\right] = \bar{a}$ and $\mathbb{E}\left[\hat{a}_{\text{DPP}}\right] = \bar{a}$ and therefore $\hat{a}_{\text{i.i.d}}$ and $\hat{a}_{\text{DPP}}$ are unbiased estimators of $\bar{a}$.

Lemma 7.1. *If $a_i \geq 0$ for all i, the estimator $\hat{a}_{DPP}$ has smaller variance than $\hat{a}_{i.i.d}$ whenever $\mathbf{K}_{ii} = p_i$ for all i.*

Proof. Since $\hat{a}_{i.i.d}$ and $\hat{a}_{\text{DPP}}$ are unbiased, it is enough to compare the second moments of the said estimators.

$$\begin{aligned}
\mathbb{E}\left[\hat{a}_{\text{DPP}}^2\right] &= \mathbb{E}\left[\sum_{ij} \frac{a_i a_j \epsilon_i \epsilon_j}{p_i p_j}\right] \\
&= \sum_{i,j} \frac{\mathbb{E}\left[\epsilon_i \epsilon_j\right] a_i a_j}{p_i p_j} \\
&= \sum_{i,j} \frac{(\mathbf{K}_{ii}\mathbf{K}_{jj} - \mathbf{K}_{ij}^2) a_i a_j}{p_i p_j} \\
&= \mathbb{E}\left[\hat{a}_{i.i.d.}^2\right] - \sum_{i \neq j} \frac{\mathbf{K}_{i,j}^2 a_i a_j}{p_i p_j} \\
&\leq \mathbb{E}\left[\hat{a}_{i.i.d.}^2\right]
\end{aligned}$$

The last inequality holds whenever $a_i \geq 0$ for all i.

□

We can also show that under appropriate conditions there exists a kernel matrix $\mathbf{K}$ such that $\mathrm{Var}(\hat{a}_{\mathrm{i.i.d}}) > \mathrm{Var}(\hat{a}_{\mathrm{DPP}})$ such that the inequality is strict.

Lemma 7.2. *If $n \geq 3$, $p_i > 0$ for all $i$and there exists i such that $p_i < 1$, then there exists a matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$ defining a DPP over - not necessarily nonnegative- $a_1, \cdots, a_n \in \mathbb{R}$ satisfying $\mathbf{K}_{i,i} = p_i$ and such that $\mathrm{Var}(\hat{a}_{\mathrm{i.i.d.}}) > \mathrm{Var}(\hat{a}_{\mathrm{DPP}})$.*

Proof. Let $\mathbf{K}$ be a matrix defining a DPP with $\mathbf{K}_{i,i} = p_i$ for all i Following the exact same proof as in Lemma 7.1, we conclude that $\mathrm{Var}(\hat{a}_{\mathrm{i.i.d}}) > \mathrm{Var}(\hat{a}_{\mathrm{DPP}})$ iff:

$$\sum_{i \neq j} \frac{\mathbf{K}^2_{i,j} a_i a_j}{p_i p_j} > 0 \tag{12}$$

We show the existence of a kernel matrix $\mathbf{K}$ for which the inequality 12 holds and $\mathbf{K}_{i,i} = p_i$ for all i.

Indeed, let $\mathbf{K} \in \mathbb{R}^{n \times n}$ be such that:

$$\mathbf{K}_{i,j} = \begin{cases} p_i & \text{if } i = j \\ \epsilon & \text{if } \frac{a_i a_j}{p_i p_j} \geq 0 \\ 0 & \text{o.w.} \end{cases}$$

For some $\epsilon > 0$. Under this definition, notice that $\sum_{i,j} \frac{\mathbf{K}^2_{i,j} a_i a_j}{p_i p_j} > 0$ and notice that since $0 \prec \mathrm{diag}(p_i) \prec I$, there exists a choice of $\epsilon > 0$ such that $0 \prec \mathbf{K} \prec \mathbb{I}_d$, thus defining a valid DPP kernel matrix $\mathbf{K}$.

□

7.1.2 Towards variance reduction for vector estimators using DPPs.

In this section we extend the results of the previous section to the multi dimensional case of Monte Carlo gradient estimators. We start with an auxiliary lemma that will be used in the variance reduction Theorems of the following sections. The following Lemma characterizes the maximum number of vectors that can all be pairwise negatively correlated. This Lemma will be used later on to argue the existence of a DPP kernel $\mathbf{K}$ for which its subsampling estimator of the Evolution Strategies gradient estimator achieves less variance than the i.i.d. subsampling estimator.

Lemma 7.3. *Let $\mathbf{v}^1, \cdots, \mathbf{v}^M \in \mathbb{R}^d$ vectors such that $\langle \mathbf{v}^i, \mathbf{v}^j \rangle < 0$ for all $i \neq j$. Then $M \leq d + 1$.*

Proof. We proceed with a proof by contradiction. Let's assume $M \geq d + 2$. Let $\mathbf{v}^1, \cdots, \mathbf{v}^{d+1}$ be a subset of $d + 1$ vectors of $\{\mathbf{v}^j\}_{j=1}^M$. There exist $a_1, \cdots, a_{d+1} \in \mathbb{R}$ such that:

$$\sum_{i=1}^{d+1} a_i \mathbf{v}^i = 0$$

If $a_i \geq 0$ for all i then $\langle \mathbf{v}^{d+2}, \sum_i a_i \mathbf{v}^i \rangle = \sum_i a_i \langle \mathbf{v}^{d+2}, \mathbf{v}^i \rangle < 0$ which would result in a contradiction. If a_i are not all nonnegative, there exist disjoint subsets $I \subset [d+2]$ and $K \subset [d+2]$ such that $I \cup J = [d+2]$, and $I \cap J = \emptyset$ and $I, J \neq \emptyset$ and with $a_i \geq 0$ for all $i \in I$ (with at least one $a_i > 0$) and $a_j \leq 0$ (with at least one $a_j < 0$) for all $j \in J$ such that:

$$\underbrace{\sum_{i \in I} a_i \mathbf{v}^i}_{I} = \underbrace{\sum_{j \in J} -a_j \mathbf{v}^j}_{II}$$

Therefore by assumption $\langle I, II \rangle < 0$ which would cause a contradiction since $I = II$.

□

Recall the gradient estimator corresponding to Evolution Strategies. If $f : \mathbb{R}^d \to \mathbb{R}$, the ES gradient estimator $\nabla f_\sigma(\theta)$ at θ equals:

$$\nabla f_\sigma(\theta) = \mathbb{E}_{\mathbf{v}\sim\mathcal{N}(0,\mathbb{I}_d)}\left[\frac{1}{\sigma} f(\theta + \sigma\mathbf{v})\mathbf{v}\right]$$

We denote by $\hat{\nabla} f_\sigma(\theta) = \frac{1}{n\sigma}\sum_{i=1}^{n} f(\theta + \sigma\mathbf{v}^i)\mathbf{v}^i$ where $\mathbf{v}^i$ are all samples from a standard Gaussian $\mathcal{N}(0, \mathbb{I}_d)$.

7.1.3 Subsampling strategies in ES

In this section we consider subsampling strategies for Evolution strategies when we have a dictionary of N sensing directions $\{\mathbf{v}^i\}_{i=1}^N$. Let $\{p_i\}_{i=1}^N$ be the ensemble of probabilities with which to sample (according to a Bernoulli trial with probability p_i) each sensing i.

We recognize two cases:

1. **Unbiased sampling** In this case we consider a subsampled-importance sampling weighted version of the empirical estimator $\hat{\nabla} f_\sigma(\theta) = \frac{1}{\sigma N}\sum_{i=1}^{N} \mathbf{v}^i f(\theta + \sigma\mathbf{v}^i)$ of the form $\hat{\nabla}_U f_\sigma(\theta) = \frac{1}{N\sigma}\sum_{i=1}^{N} \frac{\epsilon_i}{p_i}\mathbf{v}^i f(\theta + \sigma\mathbf{v}^i)$.
2. **Biased** In this case we consider a subsampled version of the empirical estimator $\hat{\nabla} f_\sigma(\theta)$ of the form $\hat{\nabla}_B f_\sigma(\theta) = \frac{1}{\sigma N}\sum_{i=1}^{N} \frac{\epsilon_i}{w_i}\mathbf{v}^i f(\theta + \sigma\mathbf{v}^i)$ where $\{w_i\}_{i=1}^N$ is a set of importance weights, not necessarily equal to $\{p_i\}$.

The crucial observation behind these estimators is that the evaluation of f need not be performed at the points that are not subsampled. This allows us to trade off computation with variance (or mean squared error). We would like to achieve the optimal tradeoff.

Unbiased subsampling

The goal of this section is to show that for any i.i.d. subsampling strategy to build an unbiased estimator for the ES gradient, there exists a DPP kernel such that the DPP unbiased subsampling estimator achieves less variance than the i.i.d. one.

The main result of this section, Theorem 7.4 concerns the estimation of functions of the form $F : \mathbb{R}^d \to \mathbb{R}^m$ as defined in Section 1, and shows that for any fixed subsampling i.i.d. strategy (encoded by subsampling probabilities $\{p_i\}$), there exists a marginal kernel $\mathbf{K}$ whose corresponding estimator achieves the same mean but has (strictly) less variance. We prove Theorem 7.5 which specializes Theorem 7.4 to the case of ES gradients. A simple notational change would render the proof valid for Theorem 7.4.

The following corresponds to Theorem 4.1 in the main text.

Theorem 7.4. *If $N \geq d + 2$ and $p_i < 1$ for all i, there exists a Marginal Kernel $\mathbf{K} \in \mathbb{R}^{N\times N}$ such that:*

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

And:

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

We show the corresponding result for the case when $\mathbb{F} = \nabla f_\sigma(\theta)$. The proof is exactly the same as in the case when considering any other type of function $F : \mathbb{R}^d \to \mathbb{R}^m$ defined as in Section 1.

Let $\mathbf{K}$ be a marginal kernel matrix defining a DPP whose samples we index as $(\epsilon_1, \cdots, \epsilon_N)$ with $\epsilon_i \in \{0, 1\}$ and such that the ensemble follows the DPP process. We consider the following subsampled ES estimator:

$$\hat{\nabla}_U^{DPP} f_\sigma(\theta) = \frac{1}{N\sigma} \sum_{i \in S} \frac{\epsilon_i}{p_i} f(\theta + \sigma \mathbf{v}^i) \mathbf{v}^i$$

Theorem 7.5. *There exists a marginal kernel* $\mathbf{K} \in \mathbb{R}^{N \times N}$ *such that* $\widehat{\mathrm{MSE}}(\hat{\nabla}_{\mathrm{U}}^{\mathrm{DPP}} f_\sigma(\theta)) < \widehat{\mathrm{MSE}}(\hat{\nabla}_{\mathrm{U}} f_\sigma(\theta))$

Proof. Since $\mathbb{E}\left[\hat{\nabla}_U^{\mathrm{DPP}} f_\sigma(\theta)\right] = \mathbb{E}\left[\hat{\nabla}_{\mathrm{U}} f_\sigma(\theta)\right]$, it is enough to show the desired statement for the square norms of these vectors.

$$\begin{aligned}
\|\hat{\nabla}_{\mathrm{U}}^{\mathrm{DPP}} f_\sigma(\theta)\|^2 &= \sum_{j=1}^{d} \left(\frac{1}{\sigma N} \sum_{i \in S} \frac{\epsilon_i}{p_i} f(\theta + \sigma \mathbf{v}^i) \mathbf{v}^i(j) \right)^2 \\
&= \frac{1}{\sigma^2 N^2} \sum_{j=1}^{d} \left(\sum_{i \in S} \frac{\epsilon_i}{p_i} f(\theta + \sigma \mathbf{v}^i) \mathbf{v}^i(j) \right)^2 \\
&= \frac{1}{\sigma^2 N^2} \sum_{j=1}^{d} \left(\sum_{i,k \in S} \frac{\epsilon_i \epsilon_k}{p_i p_k} f(\theta + \sigma \mathbf{v}^i) f(\theta + \sigma \mathbf{v}^k) \mathbf{v}^i(j) \mathbf{v}^k(j) \right)
\end{aligned}$$

Therefore:

$$\begin{aligned}
\mathbb{E}\left[\|\hat{\nabla}_{\mathrm{U}}^{\mathrm{DPP}} f_\sigma(\theta)\|^2\right] &= \mathbb{E}\left[\frac{1}{\sigma^2 N^2} \sum_{j=1}^{d} \left(\sum_{i,k \in \mathcal{S}} \frac{\epsilon_i \epsilon_k}{p_i p_k} f(\theta + \sigma \mathbf{v}^i) f(\theta + \sigma \mathbf{v}^k) \mathbf{v}^i(j) \mathbf{v}^k(j) \right) \right] \\
&= \frac{1}{\sigma^2 N^2} \sum_{j=1}^{d} \left(\sum_{i,k} \frac{\mathbb{E}[\epsilon_i \epsilon_k]}{p_i p_k} f(\theta + \sigma \mathbf{v}^i) f(\theta + \sigma \mathbf{v}^k) \mathbf{v}^i(j) \mathbf{v}^k(j) \right) \\
&= \frac{1}{\sigma^2 N^2} \sum_{j=1}^{d} \left(\sum_{i \neq k} \frac{\mathbf{K}_{i,i} \mathbf{K}_{k,k} - \mathbf{K}_{i,k}^2}{p_i p_k} f(\theta + \sigma \mathbf{v}^i) f(\theta + \sigma \mathbf{v}^k) \mathbf{v}^i(j) \mathbf{v}^k(j) \right) + \\
&\frac{1}{\sigma^2 N^2} \sum_{j=1}^{d} \left(\sum_{i=1}^{N} \frac{\mathbf{K}_{i,i}}{p_i^2} f^2(\theta + \sigma \mathbf{v}^i) \left(\mathbf{v}^i\right)^2 (j) \right)
\end{aligned}$$

Let $K_{i,i} = p_i$ for all i. The expression above becomes:

$$\begin{aligned}
\mathbb{E}\left[\|\hat{\nabla}_{\mathrm{U}}^{\mathrm{DPP}} f_\sigma(\theta)\|^2\right] &= \mathbb{E}\left[\|\hat{\nabla}_{\mathrm{U}} f_\sigma(\theta)\|^2\right] - \frac{1}{\sigma^2 N^2} \sum_{j=1}^{d} \left(\sum_{i \neq k} \frac{\mathbf{K}_{i,k}^2}{p_i p_k} f(\theta + \sigma \mathbf{v}_i) f(\theta + \sigma \mathbf{v}_k) \mathbf{v}_i(j) \mathbf{v}_k(j) \right) \\
&= \mathbb{E}\left[\|\hat{\nabla}_{\mathrm{U}} f_\sigma(\theta)\|^2\right] - \frac{1}{\sigma^2 N^2} \sum_{i \neq k} \frac{\mathbf{K}_{i,k}^2}{p_i p_k} f(\theta + \sigma \mathbf{v}^i) f(\theta + \sigma \mathbf{v}^j) \left(\sum_j \mathbf{v}^i(j) \mathbf{v}^k(j) \right) \\
&= \mathbb{E}\left[\|\hat{\nabla}_{\mathrm{U}} f_\sigma(\theta)\|^2\right] - \underbrace{\frac{1}{\sigma^2 N^2} \sum_{i \neq k} \frac{\mathbf{K}_{i,k}^2}{p_i p_k} f(\theta + \sigma \mathbf{v}^i) f(\theta + \sigma \mathbf{v}^j) \langle \mathbf{v}^i, \mathbf{v}^k \rangle}_{I}
\end{aligned}$$

Let $\mathbf{V} \in \mathbb{R}^{d \times N}$ where the $i-$th column of $\mathbf{V}$ equals $\mathbf{v}^i$, and let $\mathbf{D} \in \mathbb{R}^{N \times N}$ a diagonal matrix such that $\mathbf{D}_{i,i} = \frac{f(\theta + \sigma \mathbf{v}^i)}{p_i \sigma N}$. Let $\mathbf{K}^0 \in \mathbb{R}^{N \times N}$ be a matrix having zero diagonal entries and such that $\mathbf{K}_{i,j}^0 = \mathbf{K}_{i,j}$ with $i \neq j$. Similarly to the proof of Lemma 7.2, let's focus on term I.

$$\frac{1}{\sigma^2 N^2}\sum_{i\neq k}\frac{\mathbf{K}^2_{i,k}}{p_i p_k}f(\theta+\sigma\mathbf{v}^i)f(\theta+\sigma\mathbf{v}^j)\langle\mathbf{v}^i,\mathbf{v}^k\rangle = \sum_{i\neq k}\frac{\mathbf{K}^2_{i,k}}{\sigma^2 N^2 p_i p_k}f(\theta+\sigma\mathbf{v}^i)f(\theta+\sigma\mathbf{v}^j)\langle\mathbf{v}^i,\mathbf{v}^k\rangle$$
$$= \langle\left(\mathbf{K}^0\right)^2, \mathbf{D}^\top\mathbf{V}^\top\mathbf{D}\mathbf{V}\rangle$$

We denote by $\left(\mathbf{K}^0\right)^2$ be the matrix $\mathbf{K}^0$ with entries squared. Where $\langle\left(\mathbf{K}^0\right)^2, \mathbf{D}^\top\mathbf{V}^\top\mathbf{D}\mathbf{V}\rangle = \text{trace}(\left(\mathbf{K}^0\right)^2\mathbf{D}^\top\mathbf{V}^\top\mathbf{D}\mathbf{V})$. Define $\mathbf{K}^0$ in this way, for $i \neq j$. Let $\epsilon > 0$:

$$(\mathbf{K}^0)_{i,j} = \begin{cases}\epsilon & \text{if } f(\theta+\sigma\mathbf{v}^i)f(\theta+\sigma\mathbf{v}^j)\langle\mathbf{v}^i,\mathbf{v}^k\rangle > 0\\ 0 & \text{o.w.}\end{cases}$$

Let $\mathbf{VD}$ be the matrix with columns equal to $\mathbf{v}^i f(\theta+\sigma\mathbf{v}^i)$ and define $\mathbf{W} = \mathbf{VD}$. Consider $\mathbf{J} = \mathbf{W}^\top\mathbf{W}$ and define $\mathbf{J}^0$ be the matrix $\mathbf{J}$ without its diagonal entries. Since $N \geq d+2$, Lemma 7.3 there must be at least two positive non diagonal entries of J and therefore in this case $\langle\left(\mathbf{K}^0\right)^2, \mathbf{D}^\top\mathbf{V}^\top\mathbf{D}\mathbf{V}\rangle > 0$.

If $\mathbf{K}_{i,i} = p_i < 1$ for all i then following an argument similar to the proof of 7.2, we conclude there exists $\epsilon > 0$ such that $0 \prec \mathbf{K} \prec \mathbb{I}_d$ such that $\widehat{\text{MSE}}(\hat{\nabla}^{\text{DPP}}_{\text{U}} f_\sigma(\theta)) < \widehat{\text{MSE}}(\hat{\nabla}_{\text{U}} f_\sigma(\theta))$ as desired.

□

Theorem 7.4, yields the following corollary (corresponding to Corollary 4.1 in the main text). Under i.i.d. uniform sampling ($p_i = p$ for all i):

Corollary 7.1. *Let $\mathbf{v}^1, \cdots, \mathbf{v}^N \sim \mathcal{N}(0, I_d)$ be normally distributed sensings sampled i.i.d. Let $\hat{\nabla}_U f_\sigma(\theta)$ and $\hat{\nabla}^{DPP}_U f_\sigma(\theta)$ be subsampled gradients with $p_i = p < 1$ for all i where $\hat{\nabla}^{DPP}_U f_\sigma(\theta)$ is produced with a kernel as in Theorem 4.1. The following hold:*

$$\mathbb{E}\left[\hat{\nabla}^{DPP}_U f_\sigma(\theta)\right] = \mathbb{E}\left[\hat{\nabla}_U f_\sigma(\theta)\right] = \nabla f_\sigma(\theta)$$

And:

$$\widehat{\text{MSE}}(\hat{\nabla}^{DPP}_U f_\sigma(\theta)) < \widehat{\text{MSE}}(\hat{\nabla}_U f_\sigma(\theta))$$

This corollary implies that picking the right Kernel, subsampling perturbations from a DPP process when these perturbations are all i.i.d. Gaussian vectors, yields an unbiased estimator of the smoothed gradient $\nabla f_\sigma(\theta)$ with less variance (in this case equal to the mean squared error) than a naive subsampled gradient estimator that subsamples the $\{\mathbf{v}^i\}$ perturbations each with probability p.

Biased subsampling

The goal of this section is to show that for any i.i.d. subsampling strategy to build a biased estimator for the ES gradient, there exists a DPP kernel such that the DPP unbiased subsampling estimator achieves less mean squared error (MSE) than the i.i.d. one.

Define the biased downsampled estimator as:

$$\hat{F}(\theta)_B = \frac{1}{N}\sum_{i=1}^N \frac{\epsilon_i}{w_i} h_\theta(\mathbf{v}^i). \tag{13}$$

Theorem 4.1 and Corollary 4.1 of the main text can be generalized to the case of biased estimators. Borrowing notation from the previous section, and assuming access to an ensemble $\{w_i\}$ of importance weights, we get as a biased equivalent version of Theorem 4.1:

Theorem 7.6. *If $N \geq d+2$ and $p_i < 1$ there exists a Marginal Kernel $\mathbf{K} \in \mathbb{R}^{N\times N}$ such that:*

$$\mathbb{E}_{\{\epsilon_i\}\sim DPP(\mathbf{K})}\left[\hat{F}(\theta)^{\text{DPP}}_B\right] = \mathbb{E}_{\{\epsilon_i\}\sim\{Ber(p_i)\}}\left[\hat{F}(\theta)^{\text{iid}}_B\right],$$

and furthermore $\text{MSE}(\hat{F}(\theta)^{\text{DPP}}_B) \leq \text{MSE}(\hat{F}(\theta)^{\text{iid}}_B)$, where the comparison mean equals $\mu = \hat{F}(\theta) = \frac{1}{N}\sum_{i=1}^N h_\theta(\mathbf{v}^i)$.

Proof. The following equalities hold:

$$\mathrm{MSE}(\hat{F}(\theta)_B^{\mathrm{DPP}}) = \mathrm{Var}(\hat{F}(\theta)_B^{\mathrm{DPP}}) + \left\|\mathbb{E}\left[\hat{F}(\theta)_B^{\mathrm{DPP}} - \hat{F}(\theta)\right]\right\|^2$$

$$\mathrm{MSE}(\hat{F}(\theta)_B^{\mathrm{iid}}) = \mathrm{Var}(\hat{F}(\theta)_B^{\mathrm{iid}}) + \left\|\mathbb{E}\left[\hat{F}(\theta)_B^{\mathrm{iid}} - \hat{F}(\theta)\right]\right\|^2$$

Since the expectations of $\hat{F}(\theta)_B^{\mathrm{iid}}$ and $\hat{F}(\theta)_B^{\mathrm{DPP}}$ agree, and as a consequence of Theorem 4.1, we can produce a kernel $\mathbf{K}$ such that:

$$\mathrm{Var}(\hat{F}(\theta)_B^{\mathrm{DPP}}) < \mathrm{Var}(\hat{F}(\theta)_B^{\mathrm{iid}}),$$

the result follows. □

As a consequence of Theorem 7.6, the biased downsampled versions $\hat{\nabla}_B^{\mathrm{iid}} f_\sigma(\theta)$ and $\hat{\nabla}_B^{\mathrm{DPP}} f_\sigma(\theta)$ of the ES gradient estimator $\nabla f_\sigma(\theta)$ satisfy an analogous version of Corollary 4.1 where Var is substituted by MSE.

The proofs of Theorem 7.4 and 7.6 can be used to produce an algorithm to find kernel matrix $\mathbf{K}$ reducing MSE. The results of the previous section can be extended to the case of biased sampling estimators. These result from the case when the importance weights are different from p_i.

Similarly to the previous section, the following theorem holds. Defining $\hat{\nabla}_{\mathrm{B}} f_\sigma(\theta)$ and $\hat{\nabla}_{\mathrm{B}}^{\mathrm{DPP}} f_\sigma(\theta)$ as:

1. $\hat{\nabla}_{\mathrm{B}} f_\sigma(\theta) = \frac{1}{\sigma N} \sum_{i=1}^N \frac{\epsilon_i}{w_i} \mathbf{v}^i f(\theta + \sigma \mathbf{v}^i)$ where $\{w_i\}_{i=1}^N$ is a set of importance weights and $\epsilon_i \sim \mathrm{Ber}(p_i)$ for some probabilities ensemble $\{p_i\}$

2. $\hat{\nabla}_{\mathrm{B}}^{\mathrm{DPP}} f_\sigma(\theta) = \frac{1}{N\sigma} \sum_{i \in \mathcal{S}} \frac{\epsilon}{w_i} f(\theta + \sigma \mathbf{v}^i) \mathbf{v}^i$.

In this case, the corresponding version of Theorem 7.4 is:

Theorem 7.7. *There exists a marginal kernel $\mathbf{K} \in \mathbb{R}^{N \times N}$ such that $\widehat{\mathrm{MSE}}(\hat{\nabla}_{\mathrm{B}}^{\mathrm{DPP}} f_\sigma(\theta)) < \widehat{\mathrm{MSE}}(\hat{\nabla}_{\mathrm{B}} f_\sigma(\theta))$.*

Proof. The mean squared errors $\widehat{\mathrm{MSE}}(\hat{\nabla}_{\mathrm{B}}^{\mathrm{DPP}} f_\sigma(\theta))$ and $\widehat{\mathrm{MSE}}(\hat{\nabla}_{\mathrm{B}} f_\sigma(\theta))$ can be written as:

$$\widehat{\mathrm{MSE}}(\hat{\nabla}_{\mathrm{B}}^{\mathrm{DPP}} f_\sigma(\theta)) = \mathrm{Var}(\hat{\nabla}_{\mathrm{B}}^{\mathrm{DPP}} f_\sigma(\theta)) + \underbrace{\left\|\mathbb{E}\left[\hat{\nabla}_{\mathrm{B}}^{\mathrm{DPP}} f_\sigma(\theta))\right] - \nabla f_\sigma(\theta)\right\|^2}_{I}$$

$$\widehat{\mathrm{MSE}}(\hat{\nabla}_{\mathrm{B}} f_\sigma(\theta)) = \mathrm{Var}(\hat{\nabla}_{\mathrm{B}} f_\sigma(\theta)) + \underbrace{\left\|\mathbb{E}\left[\hat{\nabla}_{\mathrm{B}} f_\sigma(\theta))\right] - \nabla f_\sigma(\theta)\right\|^2}_{II}$$

The bias terms I and II are always equal since $\mathbb{E}\left[\hat{\nabla}_{\mathrm{B}} f_\sigma(\theta))\right] = \mathbb{E}\left[\hat{\nabla}_{\mathrm{B}}^{\mathrm{DPP}} f_\sigma(\theta))\right]$.

The remainder of the proof is exactly the same as in Theorem 4.1.

□

7.2 DPP Connections with orthogonality

In this section we flesh out some connections between structured sampling via DPPs and structured sampling via orthogonal directions such as in [33]. We show that in some way DPP structured sampling subsumes orthogonal sampling. We start showing Lemma 7.8, leading to Theorem 7.9, (Theorem 4.2 in the main text).

In what follows assume $\mathcal{X} = \{\mathbf{x}^1, \cdots, \mathbf{x}^N\}$ with $\mathbf{x}^i \in \mathbb{R}^d$ and let $\phi : \mathbb{R}^d \to \mathbb{R}^D$ be a possibly infinite feature map ϕ.

Lemma 7.8. *Let $\mathbf{W} \in \mathbb{R}^{N \times N}$ such that $\mathbf{W}_{i,j} = \langle \phi(\mathbf{x}^i), \phi(\mathbf{x}^j) \rangle$ for some a $D-$dimensional feature map ϕ. Let $A \subseteq [N]$. The nonzero eigenvalues of the principal minor W_A equal the nonzero eigenvalues of $\sum_{i \in A} \phi(\mathbf{x}^i)\phi^\top(\mathbf{x}^i)$.*

Proof. Let $A = \{i_1, \cdots, i_{|A|}\}$ and define $B_A = \left[\phi(\mathbf{x}^{i_1}) \cdots \phi(\mathbf{x}^{|A|})\right] \in \mathbb{R}^{D \times |A|}$. It follows immediately that:

$$\mathbf{W}_A = \mathbf{B}_A^\top \mathbf{B}_A$$

Assume the SVD decomposition of $\mathbf{B}_A = \mathbf{U}_A^\top \mathbf{D}_A \mathbf{V}_A$ with $\mathbf{U}_A \in \mathbb{R}^{D \times D}$, $\mathbf{D}_A \in \mathbb{R}^{D \times |A|}$, and $\mathbf{V}_A \in \mathbb{R}^{|A| \times |A|}$. And thus:

$$\mathbf{W}_A = \mathbf{V}_A^\top \underbrace{\mathbf{D}_A \mathbf{D}_A^\top}_{I} \mathbf{V}_A$$

Observe that:

$$\sum_{i \in A} \phi(\mathbf{x}^i)\phi^\top(\mathbf{x}^i) = \mathbf{B}_A \mathbf{B}_A^\top$$

And substituting the SVD decomposition of B_A yields:

$$\sum_{i \in A} \phi(\mathbf{x}^i)\phi^\top(\mathbf{x}^i) = \mathbf{U}_A^\top \underbrace{\mathbf{D}_A^\top \mathbf{D}_A}_{II} \mathbf{U}_A$$

Since the nonzero entries of I and II are the same, we conclude the nonzero eigenvalues of $\mathbf{W}_A$ and of $\sum_{i \in A} \phi(\mathbf{x}^i)\phi^\top(\mathbf{x}^i)$ coincide. □

We now show a relationship between orthogonality and DPPs.

Theorem 7.9. *Let $\mathbf{L} \in \mathbb{R}^{N \times N}$ be an $\mathbf{L}-$ensemble such that $\mathbf{L}_{i,j} = \langle \phi(\mathbf{x}^i), \phi(\mathbf{x}^j) \rangle$, where $\|\Phi(\mathbf{x}^i)\| = 1$ 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 $\mathcal{X}$.*

Proof. Recall that $\mathbb{P}_k \propto \det(\mathbf{L}_A)$. Observe also that, since all eigenvalues of $\mathbf{L}_A$ are nonnegative, if we assume the determinant of $\mathbf{L}_A$ to be nonnegative, by the arithmetic-geometric inequality:

$$(\det(\mathbf{L}_A))^{1/k} \leq \frac{\mathrm{tr}(\mathbf{L}_A)}{k} = 1 \tag{14}$$

Since the determinant equals the product of the eigenvalues while the trace is the sum. Equality holds iff all of the eigenvalues are equal to 1. Let A be a subset of size k such that all points are pairwise orthogonal after the map Φ, then $\det(\mathbf{L}_A = 1$. Furthermore, if $\det(\mathbf{L}_A) = 1$, then the set of points $\{\phi(\mathbf{x}^i)\}_{i \in A}$ must be orthogonal.

As a consequence of inequality 14, the equality $\det(\mathbf{L}_A) = t^k$ can only hold if all eigenvalues of $\mathbf{L}_A$ equal 1. We show this implies all the vectors must be orthogonal.

Let $A = \{i_1, \cdots, i_{|A|}\}$ and write $L_A^{(t)} = (\mathbf{B}_A)^\top \mathbf{B}_A$ where $B_A = \left[\phi(\mathbf{x}^{i_1}) \cdots \phi(\mathbf{x}^{i_{|A|}})\right]$. As a consequence of Lemma 7.8, the nonzero eigenvalues of $L_A^{(t)}$agree with the nonzero eigenvalues of $\Sigma = \sum_{i \in A} \phi(\mathbf{x}^i)\phi^\top(\mathbf{x}^i)$.

Since by assumption $\|\Sigma\| = t$, and $\|\phi(x_i)\| = 1$ for all i:

$$\phi^\top(\mathbf{x}^i)\Sigma\phi(\mathbf{x}^i) \leq 1$$

Expanding this equation by substituting the value of Σ, we get: $\phi^\top(\mathbf{x}^i)\Sigma\phi(\mathbf{x}^i) = \sum_{j\in A}\langle\Phi(x_j),\Phi(x_i)\rangle^2 \leq 1$

Since the term corresponding to $j = i$ already equals 1, the remaining terms must be zero. This finishes the proof.

This result implies that the subsets of points of size k with the largest mass are those corresponding to pairwise orthogonal ensembles. This finishes the proof.

□

8 Experiment Details

8.1 Code

Here we include some simple code to implement DPPMC using python 3.x.

```
import numpy as np
from pydpp.dpp import DPP

d = 10 # this will be the dimensionality of your problem
rho = 5 # this is a hyper-parameter
cov = np.eye(d) # this will be your nonisotropic covariance matrix
mu = np.repeat(0, d)
A = np.random.multivariate_normal(mu, cov, d * rho)

dpp = DPP(A)
dpp.compute_kernel(kernel_type = 'rbf')
idx = dpp.sample_k(d) # returning to original dimensionality, optional
A = A[idx]

# we now evaluate these samples.
```

This code is simple to include in any setting where samples are drawn from a nonisotropic distribution.

8.2 Optimal Choice of ρ

Here we demonstrate the impact of ρ by performing an ablation study using the CMA-ES experiments. In order to measure the importance of this parameter, we test the following values: $\rho = 2, 5, 10, 20$, and measure the mean performance across three seeds after 100 function evaluations.

As we can see in Figure 5, in most cases an increase in ρ leads to a monotonic improvement in performance. This however comes at an increase in computational cost, and as such it is important to consider the trade-off between the cost of evaluating the function vs. the DPPMC algorithm when choosing an optimal ρ for a given problem. In our experiments we choose $\rho = 10$ since this value is sufficient to achieve meaningful performance gains, demonstrating the effectiveness of our approach.

8.3 Reinforcement Learning Experiments

We provide details on the reinforcement learning experiments as follows.

Benchmark Environments. Reinforcement learning tasks are identified by a state space $\mathcal{S}$ and an action space $\mathcal{A}$. The benchmark environments consist of HalfCheetah-v2 ($|\mathcal{S}| = 17, |\mathcal{A}| = 6$), Swimmer-v2 ($|\mathcal{S}| = 8, |\mathcal{A}| = 2$), Reacher-v2 ($|\mathcal{S}| = 11, |\mathcal{A}| = 2$) and Walker2d-v2 ($|\mathcal{S}| = 17, |\mathcal{A}| = 6$). Each task takes the sensory inputs of the robot as states $s_t \in \mathcal{S}$ and motor/position controls as actions $a_t \in \mathcal{A}$. All environments are simulated via OpenAI gym [8].

Policy Architecture. We encode the policy $\pi_\theta : \mathcal{S} \mapsto \mathcal{A}$ with feed-forward network parameter θ. The architecture varies across tasks: for Swimmer-v2 and Reacher-v2, we have two hidden layers each with 16 units; for HalfCheetah-v2 and Walker2d-v2, we have two hidden layers each with 32 units. Each hidden layer is combined with a tanh non-linear function activation. The output layer

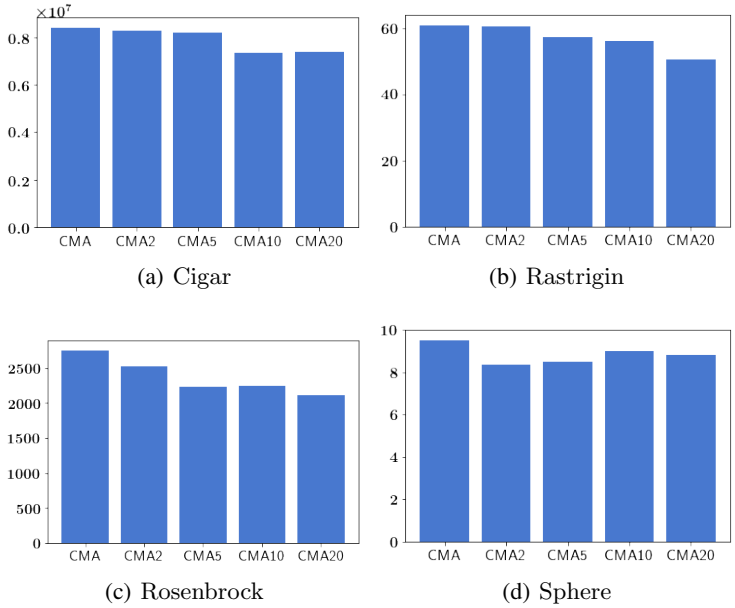
does not have non-linear function activation. For each hidden layer, instead of a fully-connected structure, we adopt a low displacement rank neural network [13] for a compact representation.

Implementations and Common Hyper-parameters. All ES algorithms are implemented with Numpy [38]. To make our implementations parallelizable, we have made heavy reference to the Ray open source project [30]. At each iteration, the ES algorithms (including Guided ES, Trust Region ES and CMA-ES) all require sampling m perturbation directions for function evaluations. We set m to be the dimension d of the policy parameter θ. Gradient based optimizations are all carried out using Adam Optimizer [20] with best learning rates chosen from $\alpha \in \{0.5, 0.1, 0.05, 0.01\}$.

DPPMC Hyper-parameters. We use a fixed RBF-kernel for all experiments: recall that a RBF-kernel takes the form $K(\mathbf{x}, \mathbf{y}) = \exp(\frac{-|\mathbf{x}-\mathbf{y}|^2}{2\sigma^2})$, we set $\sigma = 0.5$. The kernel parameter σ is manually set such that the DPPMC variants achieve good performance while the computations remain numerically stable.

Hyper-parameters for Guided ES. We follow the recipe of Guided ES [26] to set up hyper-parameters. The DPPMC variant requires constructing a sample pool of size ρm, we choose $\rho = 10$ for our experiments. The Guided ES achieves performance gains over vanilla ES by constructing non-isotropic distribution for gradient sensing, which allows for exploring subspaces where the true gradients lie. We further improve upon Guided ES with significant gains in sample efficiency.

Hyper-parameters for Trust Region ES. We follow the recipe of Trust Region ES [10] to set up hyper-parameters. Trust Region ES has two variants: (1) using ridge regression to compute update directions (Ridge); (2) using Monte-Carlo samples to estimate update directions (MC). Both variants require re-using δm samples and function evaluations from the previous iteration, here we set $\delta = 0.2$ so that the algorithm achieves $\approx 20\%$ sample gains. On top of Trust Region ES, the DPPMC variant further improves sample efficiency as demonstrated in the main paper. We refer readers to [10] for a detailed description of the algorithm.



(a) Cigar (b) Rastrigin

(c) Rosenbrock (d) Sphere

Figure 5: Comparison of CMA-ES without DPPMC vs. with DPPMC for $\rho = 2, 5, 10, 20$.