Unifying Orthogonal Monte Carlo Methods

Krzysztof Choromanski [*1] **Mark Rowland** [*2] **Wenyu Chen** [3] **Adrian Weller** [2 4]

Abstract

Many machine learning methods making use of Monte Carlo sampling in vector spaces have been shown to be improved by conditioning samples to be mutually orthogonal. Exact orthogonal coupling of samples is computationally intensive, hence approximate methods have been of great interest. In this paper, we present a unifying perspective of many approximate methods by considering Givens transformations, propose new approximate methods based on this framework, and demonstrate the first statistical guarantees for families of approximate methods in kernel approximation. We provide extensive empirical evaluations with guidance for practitioners.

1. Introduction

Monte Carlo methods are used to approximate integrals in many applications across statistics and machine learning. Back at least as far as (Metropolis & Ulam, 1949), the study of *variance reduction* or other ways to improve statistical efficiency has been a key area of research. Popular approaches include control variates, antithetic sampling, and randomized quasi-Monte Carlo (Dick & Pillichshammer, 2010).

When sampling from a multi-dimensional probability distribution, a variety of recent theoretical and empirical results have shown that coupling samples to be *orthogonal* to one another, rather than being i.i.d., can significantly improve statistical efficiency. We highlight applications in linear dimensionality reduction (Choromanski et al., 2017), locality-sensitive hashing (Andoni et al., 2015), random feature approximations to kernel methods such as Gaussian processes (Choromanski et al., 2018a) and support vector machines (Yu et al., 2016), and black-box optimization (Choromanski et al., 2018b). We refer to the class of methods using such orthogonal couplings as orthogonal Monte Carlo (OMC).

The improved statistical efficiency of OMC methods bears the cost of additional computational overhead. To reduce this cost significantly, several popular Markov chain Monte Carlo (MCMC) schemes sample from an *approximate* distribution. We refer to such schemes as approximate orthogonal Monte Carlo (AOMC). Much remains to be understood about AOMC methods, including which methods are best to use in practical settings. In this paper, we present a unifying account of AOMC methods and their associated statistical and computational considerations. In doing so, we propose several new families of AOMC methods, and provide theoretical and empirical analysis of their performance.

Our approaches are orthogonal to, and we believe could be combined with, methods in recent papers which focus on control variates (rather than couplings) for variance reduction of gradients of deep models with discrete variables (Tucker et al., 2017; Grathwohl et al., 2018).

We highlight the following novel contributions:

1. We draw together earlier approaches to scalable orthogonal Monte Carlo, and cast them in a unifying framework using the language of random Givens transformations; see Sections 2 and 3.
2. Using this framework, we introduce several new variants of approximate orthogonal Monte Carlo, which empirically have advantages over existing approaches; see Sections 3 and 4.
3. We provide a theoretical analysis of Kac's random walk, a particular AOMC method. We show that several previous theoretical guarantees for the performance of exact OMC can be extended to approximate OMC via Kac's random walk; see Section 5. In particular, to our knowledge we give the first theoretical guarantees showing that some classes of AOMCs provide gains not only in computational and space complexity, but also in accuracy, in non-linear domains (RBF kernel approximation).
4. We evaluate empirically AOMC approaches, noting relative strengths and weaknesses; see Section 6. We include an extensive analysis of the efficiency of AOMC methods in reinforcement learning evolutionary strategies, showing they can successfully replace exact OMC.

[*]Equal contribution [1]Google Brain [2]University of Cambridge [3]Massachusetts Institute of Technology [4]Alan Turing Institute. Correspondence to: Krzysztof Choromanski <kchoro@google.com>.

Proceedings of the 36th International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

2. Orthogonal Monte Carlo

Consider an expectation of the form

$$\mathbb{E}_{X\sim\mu}\left[f(X)\right] ,$$

with $\mu \in \mathscr{P}(\mathbb{R}^d)$ an isotropic probability distribution, and $f : \mathbb{R}^d \to \mathbb{R}$ a measurable, μ-integrable function. A standard Monte Carlo estimator is given by

$$\frac{1}{N}\sum_{i=1}^{N} f(X_i) , \qquad \text{where } (X_i)_{i=1}^N \overset{\text{i.i.d.}}{\sim} \mu .$$

Suppose for now that $N \leq d$. In contrast to the i.i.d. estimator above, orthogonal Monte Carlo (OMC) alters the joint distribution of the samples $(X_i)_{i=1}^N$ so that they are mutually orthogonal ($\langle X_i, X_j\rangle = 0$ for all $i \neq j$) almost-surely, whilst maintaining marginal distributions $X_i \sim \mu$ for all $i \in [N]$. As mentioned in Section 1, there are many scenarios where estimation based on OMC yields great statistical benefits over i.i.d. Monte Carlo. When $N > d$, OMC methods are extended by taking independent collections of d samples which are mutually orthogonal.

We note that for an isotropic measure $\mu \in \mathscr{P}(\mathbb{R}^d)$, in general there exist many different joint distributions for $(X_i)_{i=1}^N$ that induce an orthogonal coupling.

Example 2.1. *Let $\mu \in \mathscr{P}(\mathbb{R}^d)$ be an isotropic distribution, and let ρ_μ be the corresponding distribution of the norm of a vector with distribution μ. Let $\mathbf{v}_1, \ldots, \mathbf{v}_d$ be the rows of a random orthogonal matrix drawn from Haar measure on $\mathscr{O}(d)$ (the group of orthogonal matrices in $\mathbb{R}^{d\times d}$) and let $R_1, \ldots, R_d \overset{i.i.d.}{\sim} \rho$. Then both $(R_i\mathbf{v}_i)_{i=1}^d$ and $(R_1\mathbf{v}_i)_{i=1}^d$ form OMC sequences for μ. More advanced schemes may incorporate non-trivial couplings between the $(R_i)_{i=1}^d$.*

Example 2.1 illustrates that although a variety of OMC couplings exist for any given target distribution, all such algorithms have in common the task of sampling an exchangeable collection of mutually orthogonal vectors $\mathbf{v}_1, \ldots, \mathbf{v}_d$ such that each vector marginally has uniform distribution over the sphere S^{d-1}. We state this in an equivalent form below.

Problem 2.2. *Sample a matrix $\mathbf{M}$ from Haar measure on $\mathscr{O}(d)$, the group of orthogonal matrices in $\mathbb{R}^{d\times d}$.*

Several methods are known for solving this problem exactly (Genz, 1998; Mezzadri, 2007), involving Gram-Schmidt orthogonalisation, QR decompositions, and products of Householder and Givens rotations. Computationally, these methods incur high costs:

(i) Computational cost of sampling. All OMC methods require $\mathcal{O}(d^3)$ time to sample a matrix vs. $\mathcal{O}(d^2)$ for i.i.d. Monte Carlo.

(ii) Computational cost of computing matrix-vector products. If the matrix $\mathbf{M}$ is only required in order to compute matrix-vector products, then the Givens and Householder methods yield such products in $\mathcal{O}(d^2)$ time, without needing to construct the full matrix $\mathbf{M}$. The Gram-Schmidt method does not offer this advantage.

(iii) Space requirements. All methods require the storage of $\mathcal{O}(d^2)$ floating-point numbers.

Approximate OMC methods are motivated by the desire to reduce the computational overheads of exact OMC, whilst still maintaining statistical advantages that arise from orthogonality. Additionally, it turns out in many cases that it is simultaneously possible to improve on (ii) and (iii) in the list above, via the use of structured matrices. Indeed, we will see that good quality AOMC methods can achieve $\mathcal{O}(d^2 \log d)$ sampling complexity, $\mathcal{O}(d \log d)$ matrix-vector product complexity, and $\mathcal{O}(d)$ space requirements.

3. Approximate Orthogonal Monte Carlo

In Section 2, we saw that the sampling problem in OMC is reducible to sampling random matrices from $\mathscr{O}(d)$ according to Haar measure, and that the best known complexity for performing this task exactly is $\mathcal{O}(d^3)$. For background details on approximating Haar measure on $\mathscr{O}(d)$, see reviews by Genz (1998); Mezzadri (2007). Here, we review several approximate methods for this task, including Hadamard-Rademacher random matrices, which have proven popular recently, and cast them in a unifying framework. We begin by recalling the notion of a *Givens rotation* (Givens, 1958).

Definition 3.1. *A d-dimensional* Givens rotation *is an orthogonal matrix specified by two distinct indices $i, j \in [d]$, and an angle $\theta \in [0, 2\pi)$. The Givens rotation is then given by the matrix $\mathbf{G}[i,j,\theta]$ satisfying*

$$\mathbf{G}[i,j,\theta]_{k,l} = \begin{cases} \cos(\theta) & \textit{if } k = l \in \{i,j\} \\ -\sin(\theta) & \textit{if } k = i, l = j \\ \sin(\theta) & \textit{if } k = j, l = i \\ 1 & \textit{if } k = l \notin \{i,j\} \\ 0 & \textit{otherwise} . \end{cases}$$

Thus, the Givens rotation $\mathbf{G}[i,j,\theta]$ fixes all coordinates of $\mathbb{R}^d$ except i and j, and in the two-dimensional subspace spanned by the corresponding basis vectors, it performs a rotation of angle θ. A Givens rotation $\mathbf{G}[i,j,\theta]$ composed on the right with a reflection in the j coordinate will be termed a Givens reflection *and written $\widetilde{\mathbf{G}}[i,j,\theta]$. Givens rotations and reflections will be generically referred to as* Givens transformations.

We now review several popular methods for AOMC, and show that they may be understood in terms of Givens transformations.[1]

[1] We briefly note that some methods always return matrices

3.1. Kac's Random Walk

Kac's random walk composes together a series of random Givens rotations to obtain a random orthogonal matrix. It may thus be interpreted as a random walk over the special orthogonal group $\mathscr{SO}(d)$. Formally, it is defined as follows.

Definition 3.2 (Kac's random walk)**.** *Kac's random walk on $\mathscr{SO}(d)$ is defined to be the Markov chain $(\mathbf{K}_T)_{T=1}^\infty$, given by*

$$\mathbf{K}_T = \prod_{t=1}^{T} \mathbf{G}[I_t, J_t, \theta_t] \, ,$$

where for each $t \in \mathbb{N}$, the random variables $(I_t, J_t) \sim \mathrm{Unif}([d]^{(2)})$ and $\theta_t \sim \mathrm{Unif}([0, 2\pi))$ are independent.

Here and in the sequel, the product notation $\prod_{t=1}^T \mathbf{M}_t$ always denotes the product $\mathbf{M}_T \cdots \mathbf{M}_1$, with the highest-index matrix appearing on the left. It is well known that Kac's random walk is ergodic, and has Haar measure on $\mathscr{SO}(d)$, the special orthogonal group, as its unique invariant measure. More recently, finite-time analysis of Kac's random walk has established its mixing time as $\mathcal{O}(d^2 \log d)$ (Oliveira, 2009). Further, considering a fixed vector $\mathbf{v} \in S^{d-1}$, the sequence of random variables $(\mathbf{K}_t \mathbf{v})_{t=1}^\infty$ can be interpreted as a Markov chain on S^{d-1}, and it is known to converge to the uniform distribution on the sphere, with mixing time $\mathcal{O}(d \log d)$ (Pillai & Smith, 2017). Thus, an approximation to Haar measure on $\mathscr{O}(d)$ may be achieved by simulating Kac's random walk for a certain number of steps; the mixing times described above give a guide as to the number of steps required for a close approximation.

3.2. Hadamard-Rademacher Matrices

Another popular mechanism for approximating Haar measure are Hadamard-Rademacher random matrices. These involve taking products between random diagonal matrices, and certain structured deterministic Hadamard matrices.

Definition 3.3 (Hadamard-Rademacher chain)**.** *The Hadamard-Rademacher chain on $\mathscr{O}(2^L)$ is defined to be the following Markov chain $(\mathbf{X}_T)_{T=1}^\infty$, given by*

$$\mathbf{X}_T = \prod_{t=1}^{T} \mathbf{H}\mathbf{D}_t \, , \tag{1}$$

where $(\mathbf{D}_t)_{t=1}^\infty$ are independent random diagonal matrices, with each diagonal element a Rademacher ($\mathrm{Unif}(\{\pm 1\})$*) random variable, and $\mathbf{H}$ is the normalised Hadamard matrix, defined as the following Kronecker product*

$$\mathbf{H} = \underbrace{\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{pmatrix} \otimes \cdots \otimes \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{pmatrix}}_{L \text{ times}} \, .$$

These matrices $\mathbf{X}_T$ (typically with $T \in \{1, 2, 3\}$) have been used recently in the context of dimensionality reduction (Choromanski et al., 2017) (see also (Ailon & Chazelle, 2009)), kernel approximation (Yu et al., 2016), and locality-sensitive hashing (Andoni et al., 2015). Ailon & Chazelle (2009) give an interpretation of such matrices as randomised discrete Fourier transforms; here, we show that they can be thought of as products of random Givens rotations with more structure than in Kac's random walk, giving a unifying perspective on the two methods. To do this, we first require some notation. It is a classical result that the Hadamard matrix $\mathbf{H} \in \mathbb{R}^{2^L \times 2^L}$ can be understood as the discrete Fourier transform over the additive Abelian group $\mathbb{F}_2^L$, by identifying $\{1, \ldots, 2^L\}$ with $\mathbb{F}_2^L$ in the following manner. We associate the element $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_L) \in \mathbb{F}_2^L$ with the element $x \in \{1, \ldots, 2^L\}$ with the property that $x - 1$ expressed in binary is $\lambda_L \ldots \lambda_1$. With this correspondence understood, we will write expressions such as $\widetilde{\mathbf{G}}[\boldsymbol{\lambda}, \boldsymbol{\lambda}', \theta]$ for $\boldsymbol{\lambda}, \boldsymbol{\lambda}' \in \mathbb{F}_2^L$ without further comment. Denoting the canonical basis of $\mathbb{F}_2^L$ by $\mathbf{e}_1, \ldots, \mathbf{e}_L$, define, for $j \in \{1, \ldots, L\}$,

$$\widetilde{\mathbf{F}}^{j,L} = \prod_{\substack{\lambda \in \mathbb{F}_2^L \\ \lambda_j = 0}} \widetilde{\mathbf{G}}[\boldsymbol{\lambda}, \boldsymbol{\lambda} + \mathbf{e}_j, \pi/4] \in \mathscr{O}(2^L) \, . \tag{2}$$

Then the normalised Hadamard matrix $\mathbf{H}_L \in \mathscr{O}(2^L)$ can be written

$$\mathbf{H}_L = \prod_{i=1}^{L} \widetilde{\mathbf{F}}^{i,L} \, . \tag{3}$$

Thus, $\mathbf{H}_L$ is naturally described as the product of Givens reflections as above, and indeed it is this decomposition which exactly describes the operations constituting the fast Hadamard transform. These relationships are illustrated in Figure 1, with further illustration in Appendix Section D.

Thus, we may give a new interpretation of the Hadamard-Rademacher random matrix $\mathbf{HD}_t$ appearing in Expression (1), by writing

$$\mathbf{HD}_t = \left(\prod_{i=1}^{L-1} \widetilde{\mathbf{F}}^{i,L} \right) (\widetilde{\mathbf{F}}^{L,L} \mathbf{D}_t) \, .$$

In this expression, we may interpret $\widetilde{\mathbf{F}}^{L,L}\mathbf{D}_t$ as a product of random Givens transformations with a deterministic, structured choice of rotation axes, and rotation angle chosen uniformly from $\{\pi/4, -3\pi/4\}$, and chosen uniformly at random to be a rotation or reflection. This perspective will allow us to generalise this popular class of AOMC methods in Section 4.

with determinant 1 (i.e. taking values in the special orthogonal group $\mathscr{SO}(d)$); such methods are easily adjusted to yield matrices across the full orthogonal group $\mathscr{O}(d)$ by composing with diagonal matrix with $\mathrm{Unif}(\{\pm 1\})$ entries. We will not mention this in the sequel.

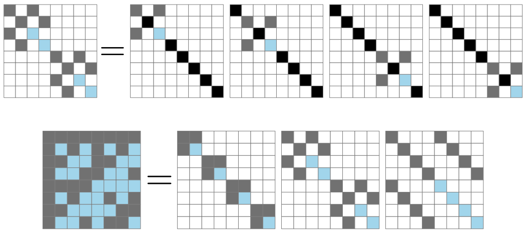


Figure 1. Top: the matrix $\widetilde{\mathbf{F}}^{2,3}$ expressed as a commuting product of Givens reflections, as in Expression (2). Bottom: the normalised Hadamard matrix $\mathbf{H}_3$ written as a product of $\widetilde{\mathbf{F}}^{1,3}$, $\widetilde{\mathbf{F}}^{2,3}$ and $\widetilde{\mathbf{F}}^{3,3}$. Matrix elements are coloured white/black to represent 0/1 elements, and grey/blue to represent elements in $(0,1)$ and $(-1,0)$.

3.3. Butterfly Matrices

Butterfly matrices generalise Hadamard-Rademacher random matrices and are a well known means of approximately sampling from Haar measure. They have found recent application in random feature sampling for kernel approximation (Munkhoeva et al., 2018). A butterfly matrix is given by defining transform matrices of the form

$$\mathbf{F}^{j,L}[(\theta_{j,\boldsymbol{\mu}})_{\boldsymbol{\mu}\in\mathbb{F}_2^{L-j}}] = \prod_{\substack{\boldsymbol{\lambda}\in\mathbb{F}_2^L \\ \boldsymbol{\lambda}_j=0}} \mathbf{G}[\boldsymbol{\lambda}, \boldsymbol{\lambda}+\mathbf{e}_j, \theta_{j,\boldsymbol{\lambda}_{j+1:L}}] \in \mathscr{O}(2^L)\,.$$

Then the butterfly matrix $\mathbf{B}_L$ is the random matrix taking values in the special orthogonal group $\mathscr{S}\mathscr{O}(2^L)$ as below, where $((\theta_{i,\boldsymbol{\mu}})_{\boldsymbol{\mu}\in\mathbb{F}_2^{L-i}})_{i=1}^L \overset{\text{i.i.d.}}{\sim} \text{Unif}([0,2\pi))$:

$$\mathbf{B}_L = \prod_{i=1}^{L} \mathbf{F}^{i,L}[(\theta_{i,\boldsymbol{\mu}})_{\boldsymbol{\mu}\in\mathbb{F}_2^{L-i}}]\,. \tag{4}$$

Thus butterfly matrices and Hadamard-Rademacher matrices may both be viewed as 'versions' of Kac's random walk that introduce statistical dependence between various random variables.

4. New AOMC Methods

Having developed a unifying perspective of existing AOMC methods in terms of Givens rotations, we now introduce two new families of AOMC methods that extend this framework.

4.1. Structured Givens Products

We highlight the work of Mathieu & LeCun (2014), who propose to (approximately) parametrise $\mathscr{O}(2^L)$ as a structured product of Givens rotations, for the purposes of learning approximate factorised Hessian matrices. This construction is straightforward to randomise, and yields a new method for AOMC, generalising both Hadamard-Rademacher random matrices and butterfly random matrices, defined precisely as:

$$\prod_{j=1}^{L}\left[\prod_{\substack{\boldsymbol{\lambda}\in\mathbb{F}_2^L \\ \boldsymbol{\lambda}_j=0}} \mathbf{G}[\boldsymbol{\lambda}, \boldsymbol{\lambda}+\mathbf{e}_j, \theta_{i,\boldsymbol{\lambda}}]\right]\,,$$

where $(\theta_{i,\boldsymbol{\lambda}})_{\boldsymbol{\lambda}\in\mathbb{F}_2^L, i\in[L]} \overset{\text{i.i.d.}}{\sim} \text{Unif}([0,2\pi))$. This can be understood as generalising random butterfly matrices by giving each constituent Givens rotation an independent rotation angle, whereas in Expression (4), some Givens rotations share the same random rotation angles.

4.2. Hadamard-MultiRademacher matrices

Given the representation of Hadamard-Rademacher matrices in Expression (1), a natural generalisation of these matrices is given by the notion of a Hadamard-MultiRademacher random matrix, defined below.

Definition 4.1. *The Hadamard-MultiRademacher random matrix on $\mathscr{O}(2^L)$ is defined by the product*

$$\prod_{i=1}^{L}\left(\widetilde{\mathbf{F}}^{i,L}\mathbf{D}_i\right)\,, \tag{5}$$

where $(\widetilde{\mathbf{F}}^{i,L})_{i=1}^L$ are the structured products of deterministic Givens reflections of Expression (2), and $(\mathbf{D}_i)_{i=1}^L$ are independent random diagonal matrices, with each diagonal element having independent Rademacher distribution.

5. Approximation Theory

Having described various AOMC methods and their computational advantages, we now turn to statistical properties. We consider theoretical guarantees first when AOMC methods are used for linear dimensionality reduction, and then for non-linear applications. Analysis of Hadamard-Rademacher matrices for linear dimensionality reduction was undertaken by Choromanski et al. (2017); in Section 5.1 we contribute similar analysis for Hadamard-MultiRademacher random matrices and Kac's random walk. In contrast, extending theoretical guarantees in non-linear applications (such as random feature kernel approximation) from exact OMC methods to AOMC methods has not yet been possible, to the best of our knowledge. In Section 5.2, we give the first guarantees that the statistical benefits in kernel approximation that OMC methods yield are also available when using AOMC methods based on Kac's random walk. All proofs are in the Appendix.

5.1. Linear Dimensionality Reduction Analysis

Consider the linear (dot-product) kernel defined as: $K(\mathbf{x},\mathbf{y}) = \langle\mathbf{x},\mathbf{y}\rangle$, for all $\mathbf{x},\mathbf{y}\in\mathbb{R}^d$. In the dimensionality reduction setting the goal is to find a mapping $\Psi : \mathbb{R}^d \to \mathbb{R}^m$ such that $m < d$ and $\langle \Psi(\mathbf{x}_i), \Psi(\mathbf{x}_j) \rangle \approx K(\mathbf{x}_i, \mathbf{x}_j)$ for all $i, j \in [N]$, for some dataset $\{\mathbf{x}_i\}_{i=1}^N \subseteq \mathbb{R}^d$. The *random projections* approach to this problem defines a random linear map $\Psi_m(\mathbf{x}) = \frac{\sqrt{d}}{\sqrt{m}} \mathbf{Mx}$ (for all $\mathbf{x} \in \mathbb{R}^d$), with $\mathbf{M}$ a random matrix taking values in $\mathbb{R}^{m \times d}$. A commonly used random projection is given by taking $\mathbf{M}$ to have i.i.d. $N(0, 1/d)$ entries. This yields the unstructured Johnson-Lindenstrauss transform (Johnson & Lindenstrauss, 1984, JLT), with corresponding dot-product estimator given by $\widehat{K}_m^{\text{base}}(\mathbf{x}, \mathbf{y}) = \frac{d}{m} (\mathbf{Mx})^\top (\mathbf{My})$. Several improvements on the JLT have been proposed, yielding computational benefits (Ailon & Chazelle, 2009; Dasgupta et al., 2010). In the context of AOMC methods, Choromanski et al. (2017) demonstrated that by replacing the Gaussian matrix in the Johnson-Lindenstrauss transform with a general Hadamard-Rademacher matrix composed with a random coordinate projection matrix $\mathbf{P}$ uniformly selecting m coordinates without replacement, it is possible to simultaneously improve on the standard JLT in terms of: (i) estimator MSE, (ii) cost of computing embeddings, (iii) storage space for the random projection, and (iv) cost of sampling the random projection.

We show new results that similar improvements are available for random projections based on Hadamard-MultiRademacher random matrices and Kac's random walk – specifically, projections of the form

$$\Psi_m^{\text{HMD}}, \Psi_{k,m}^{\text{KAC}} : \mathbf{x} \mapsto \frac{\sqrt{d}}{\sqrt{m}} \mathbf{PMx}\,, \quad \forall \mathbf{x} \in \mathbb{R}^d\,, \qquad (6)$$

where $\mathbf{M}$ is either a Hadamard-MultiRademacher random matrix (Definition 4.1), or a Kac's random walk matrix with k Givens rotations (Definition 3.2). We denote the corresponding dot-product estimators by $\widehat{K}_m^{\text{HMD}}(\mathbf{x}, \mathbf{y})$ and $\widehat{K}_{k,m}^{\text{KAC}}(\mathbf{x}, \mathbf{y})$, respectively.

Theorem 5.1. *The Hadamard-MultiRademacher dot-product estimator has MSE given by:*

$$\text{MSE}(\widehat{K}_m^{\text{HMD}}(\mathbf{x}, \mathbf{y})) =$$
$$\frac{1}{m} \left(\frac{d-m}{d-1} \right) \left(\|\mathbf{x}\|_2^2 \|\mathbf{y}\|_2^2 + \langle \mathbf{x}, \mathbf{y} \rangle^2 - 2 \sum_{\boldsymbol{\lambda} \in \mathbb{F}_2^L} x_{\boldsymbol{\lambda}}^2 y_{\boldsymbol{\lambda}}^2 \right) .$$

Comparing with the known formula for $\text{MSE}(\widehat{K}_m^{\text{base}}(\mathbf{x}, \mathbf{y}))$ in (Choromanski et al., 2017), the MSE associated with the Hadamard-MultiRademacher embedding is strictly lower.

Theorem 5.2. *The dot-product estimator based on Kac's random walk with k steps has MSE given by*

$$\text{MSE}(\widehat{K}_{k,m}^{\text{KAC}}(\mathbf{x}, \mathbf{y})) = \frac{d}{m} \left(\frac{d-m}{d-1} \right) \left(-\frac{\langle \mathbf{x}, \mathbf{y} \rangle^2}{d} + \chi \right) ,$$

where $\chi = \Theta^k \sum_{i=1}^d x_i^2 y_i^2 + \frac{1-\Theta^k}{2(1-\Theta)d(d-1)} (2 \langle \mathbf{x}, \mathbf{y} \rangle^2 + \|\mathbf{x}\|_2^2 \|\mathbf{y}\|_2^2)$ *and* $\Theta = \frac{(d-2)(2d+1)}{2d(d-1)}$. *In particular, there exists a universal constant $C > 0$ such that for $k = Cd \log(d)$ the following holds:*

$$\text{MSE}(\widehat{K}_{k,m}^{\text{KAC}}(\mathbf{x}, \mathbf{y})) < \text{MSE}(\widehat{K}_m^{\text{base}}(\mathbf{x}, \mathbf{y})).$$

As we see, estimators using only $\mathcal{O}(d \log d)$ Givens random rotations are more accurate than unstructured baselines and they also provide computational gains.

5.2. Non-linear Kernel Approximation Analysis

Kernel methods such as Gaussian processes and support vector machines are widely used in machine learning. Given a stationary isotropic continuous kernel $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, with $K(\mathbf{x}, \mathbf{y}) = \phi(\|\mathbf{x} - \mathbf{y}\|)$ for some positive definite function $\phi : \mathbb{R} \to \mathbb{R}$, the celebrated Bochner's theorem states that there exists a probability measure $\mu_\phi \in \mathscr{P}(\mathbb{R}^d)$ such that:

$$K^\phi(\mathbf{x}, \mathbf{y}) = \text{Re} \int_{\mathbb{R}^d} \exp(i \mathbf{w}^\top (\mathbf{x} - \mathbf{y})) \mu_\phi(\text{d}\mathbf{w})\,. \qquad (7)$$

Rahimi & Recht (2007) proposed to use a Monte Carlo approximation, yielding a random feature map $\Psi_{m,d} : \mathbb{R}^d \to \mathbb{R}^{2m}$ given by

$$\Psi_{m,d}(\mathbf{x}) = \left(\frac{1}{\sqrt{m}} \cos(\mathbf{w}_i^\top \mathbf{x}), \frac{1}{\sqrt{m}} \sin(\mathbf{w}_i^\top \mathbf{x}) \right)_{i=1}^m ,$$

with $(\mathbf{w}_i)_{i=1}^m \overset{\text{i.i.d.}}{\sim} \mu_\phi$. Inner products of these features:

$$\widehat{K}_{\text{base}}^{\phi,m}(\mathbf{x}, \mathbf{y}) = \langle \Psi_{m,d}(\mathbf{x}), \Psi_{m,d}(\mathbf{y}) \rangle \qquad (8)$$

are then standard Monte Carlo estimators of Expression (7), allowing computationally fast linear methods to be used in approximation non-linear kernel methods. Yu et al. (2016) proposed to couple the directions of the $(\mathbf{w}_i)_{i=1}^m$ to be orthogonal almost surely, whilst keeping their lengths independent. Empirically this leads to substantial empirical variance reduction, but in order for the method to be practical, an AOMC method is required to simulate the orthogonal directions; Yu et al. (2016) used Hadamard-Rademacher random matrices. However, theoretical improvements were only proven for exact OMC methods (Yu et al., 2016; Choromanski et al., 2018a); thus, the empirical success of AOMC methods in this domain were unaccounted for.

Here, we close this gap, showing that using AOMC simulation of the directions of $(\mathbf{w}_i)_{i=1}^m$ using Kac's random walk leads to provably lower-variance estimates of kernel values in Expression (7) than for the i.i.d. approach. Before stating this result formally, we introduce some notation.

Definition 5.3. *We denote by $\mathcal{GRR}_d^k$ a distribution over the orthogonal group $\mathscr{O}(d)$ corresponding to Kac's random walk with k Givens rotations.*

Definition 5.4. *For a $1D$-distribution Φ, we denote by $\mathcal{GRR}_d^{\Phi,k}$ the distribution over matrices in $\mathbb{R}^{d\times d}$ given by the distribution of the product $\mathbf{DA}$, where $\mathbf{A}\sim\mathcal{GRR}_d^k$ and independently, $\mathbf{D}$ is a diagonal matrix with diagonal entries sampled independently from Φ.*

We denote the kernel estimator using random vectors $(\mathbf{w}_i)_{i=1}^m$ drawn from $\mathcal{GRR}_d^{\Phi,k}$ (rather than i.i.d. samples from μ_ϕ) by $\widehat{K}_{\text{kac}}^{\phi,m,k}(\mathbf{x},\mathbf{y})$. We also denote by $S(\epsilon)$ a ball of radius ϵ and centered at 0. We now state our main result.

Theorem 5.5 (Kac's random walk estimators of RBF kernels)**.** *Let $K_d:\mathbb{R}^d\times\mathbb{R}^d\rightarrow\mathbb{R}$ be the Gaussian kernel and let $\epsilon>0$. Let $\mathcal{B}$ be a set satisfying $\mathrm{diam}(\mathcal{B})\leq B$ for some universal constant B that does not depend on d ($\mathcal{B}$ might be for instance a unit sphere). Then there exists a constant $C=C(B,\epsilon)>0$ such that for every $\mathbf{x},\mathbf{y}\in\mathcal{B}\backslash S(\epsilon)$ and d large enough we have:*

$$\mathrm{MSE}(\widehat{K}_{\text{kac}}^{\phi,m,k}(\mathbf{x},\mathbf{y}))<\mathrm{MSE}(\widehat{K}_{\text{base}}^{\phi,m}(\mathbf{x},\mathbf{y})),$$

where $k=C\cdot d\log d$ and $m=ld$ for some $l\in\mathbb{N}$.

Let us comment first on the condition $\mathbf{x},\mathbf{y}\in\mathcal{B}\backslash S(\epsilon)$. This is needed to avoid degenerate cases, such as $\mathbf{x}=\mathbf{y}=0$, where both MSEs are trivially the same. Separation from zero and boundedness are mild conditions and hold in most practical applications. Whilst the result is stated in terms of the Gaussian kernel, it holds more generally; results are given in the Appendix. We emphasise that, to our knowledge, this is the first result showing that AOMC methods can be applied in non-linear estimation tasks and achieve improved statistical performance to i.i.d. methods, whilst simultaneously incurring a lower computational cost, due to requiring only $\mathcal{O}(d\log d)$ Givens rotations.

We want to emphasize that we did not aim to obtain optimal constants in the above theorems. In the experimental section we show that in practice we can choose small values for them. In particular, for all experiments using Kac's random walk matrices we use $C=2$.

6. Experiments

We illustrate the theory of Section 5 with a variety of experiments, and provide additional comparisons between the AOMC methods described in Sections 3 and 4. In all experiments, we used $Cd\log(d)$ rotations with $C=2$ for the KAC mechanism. We note that there is a line of work on learning some of these structured contructions (Jing et al., 2017), but in this paper we focus on randomized transformations.

6.1. MMD Comparisons

We directly compare the distribution of $\mathbf{M}$ obtained from AOMC algorithms with Haar measure on $\mathscr{O}(d)$ via maximum mean discrepancy (MMD) (Gretton et al., 2012). Given a set $\mathcal{X}$, MMD is a distance on $\mathscr{P}(\mathcal{X})$, specified by choosing a kernel $K:\mathcal{X}\times\mathcal{X}\rightarrow\mathbb{R}$, which encodes similarities between pairs of points in $\mathcal{X}$. The squared MMD between two distributions $\eta,\mu\in\mathscr{P}(\mathcal{X})$ is then defined by

$$\begin{aligned}\mathrm{MMD}(\eta,\mu)^2=&\,\mathbb{E}_{X,X'}\left[K(X,X')\right] \\ &-2\mathbb{E}_{X,Y}\left[K(X,Y)\right]+\mathbb{E}_{Y,Y'}\left[K(Y,Y')\right]\,,\end{aligned} \tag{9}$$

where $X,X'\overset{\text{i.i.d.}}{\sim}\eta$, and independently, $Y,Y'\overset{\text{i.i.d.}}{\sim}\mu$. Many metrics can be used to compare probability distributions. MMD is a natural candidate for these experiments for several reasons: (i) it straightforward to compute unbiased estimators of the MMD given samples from the distributions concerned, unlike e.g. Wasserstein distance; (ii) MMD takes into account geometric information about the space $\mathcal{X}$, unlike e.g. total variation; and (iii) in some cases, it is possible to deal with uniform distributions analytically, rather than requiring approximation through samples.

The comparison we make is the following. For fixed vectors $\mathbf{v}\in S^{d-1}$, we compare the distribution of $\mathbf{Mv}$ against uniform measure on the sphere S^{d-1}, for cases where $\mathbf{M}$ is drawn from an AOMC method. In order to facilitate comparison of various AOMC methods, we compare number of floating-point operations (FLOPs) required to evaluate matrix-vector products vs. MMD squared between the two distributions on the sphere described above; we use FLOPs to facilitate straightforward comparison between methods without needing to consider specific implementation details and hardware optimisation, but observe that in practice, such considerations may also warrant attention.

To use the MMD metric defined in Equation (9), we require a kernel $K:S^{d-1}\times S^{d-1}\rightarrow\mathbb{R}$. We propose the exponentiated-angular kernel, defined by $K_\lambda(\mathbf{x},\mathbf{y})=\exp(-\lambda\theta(\mathbf{x},\mathbf{y}))$ for $\lambda>0$, where $\theta(\mathbf{x},\mathbf{y})$ is the angle between $\mathbf{x}$ and $\mathbf{y}$. With this kernel, we can analytically integrate out the terms in Equation (9) concerning the uniform distribution on the sphere (see Appendix for details). Results for comparing FLOPs against MMD are displayed in Figure 2. Several interesting observations can be made.

First, whilst a single Hadamard-Rademacher matrix incurs a low number of FLOPs relative to other methods (by virtue of the restriction on the angles appearing in their Givens rotation factorisations; see Section 3), this comes at a cost of significantly higher squared MMD relative to competing methods. Pleasingly, the Hadamard-MultiRademacher random matrix achieves a much more competitive squared MMD without incurring any additional FLOPs, making this newly-proposed method a strong contender as judged by an MMD vs. FLOPs trade-off. Secondly, butterfly and structured Givens product matrices incur higher numbers of FLOPs due to the lack of restrictions placed on the random angles in their Givens factorisations, but achieve extremely

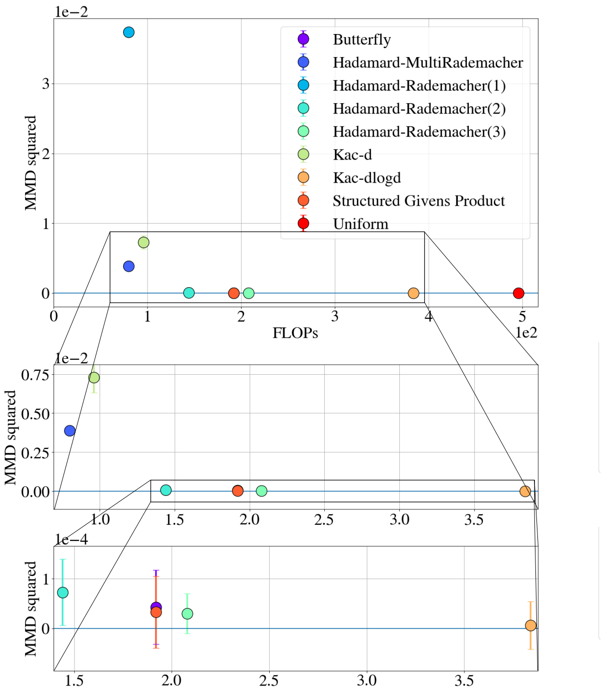


Figure 2. MMD squared vs. floating-point operations required for matrix-vector products, dimensionality 16.

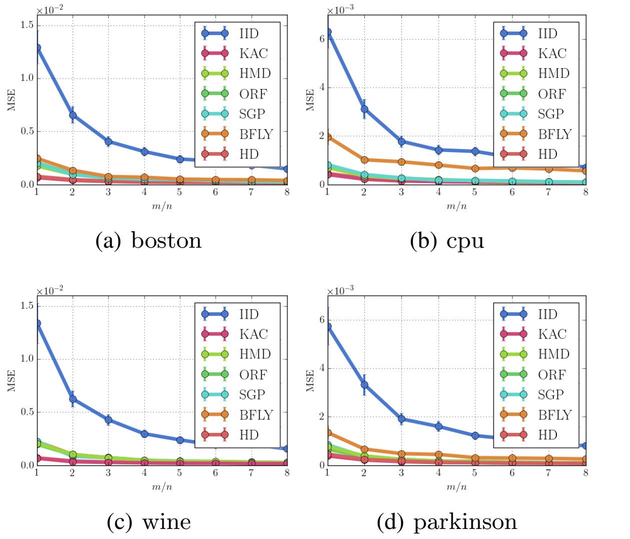
small squared MMD. Finally, we observe the dramatic savings in FLOPs that can be made, even in modest dimensions, by passing from exact OMC methods to AOMC methods.

6.2. Kernel Approximation

We present experiments on four datasets: boston, cpu, wine, parkinson (more datasets studied in the Appendix).

Pointwise kernel approximation: We computed empirical mean squared error (MSE) for several estimators of a Gaussian kernel and dot-product kernel considered in this paper for several datasets (see Appendix). We tested the following estimators: baseline using Gaussian unstructured matrices (IID), exact OMC using Gaussian orthogonal matrices and producing orthogonal random features (ORF), AOMC methods using Hadamard-Rademacher matrices (HD) with three HD blocks, Hadamard-MultiRademacher matrices (HMD), Kac's random walk matrices (KAC), structured Givens products (SGP), and butterfly matrices (BFLY). Results for the Gaussian kernel are presented in Fig. 3, 4.

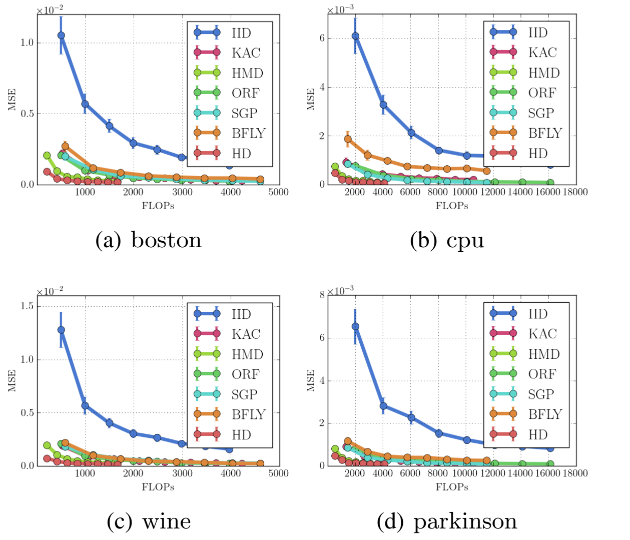
Approximating kernel matrices: We test the relative error of kernel matrix estimation for the above estimators for the Gaussian kernel (following the setting of Choromanski & Sindhwani, 2016). Results are presented in Figure 5.



(a) boston (b) cpu

(c) wine (d) parkinson

Figure 3. Empirical MSE (mean squared error) for the pointwise evaluation of the Gaussian kernel for different MC estimators.



(a) boston (b) cpu

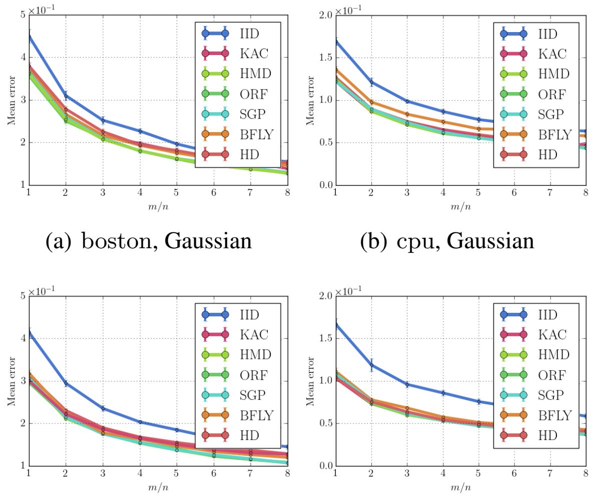
(c) wine (d) parkinson

Figure 4. Number of FLOPs required to reach particular empirical MSE levels for the pointwise evaluation of the Gaussian kernel for different MC estimators.

6.3. Policy Search

We consider here applying proposed classes of structured matrices to construct AOMCs for the gradients of Gaussian smoothings of blackbox functions that can be used for blackbox optimization. The *Gaussian smoothing* (Nesterov & Spokoiny, 2017) of a blackbox function F is given as:

$$F_\sigma(\theta) = \mathbb{E}_{\mathbf{g}\sim\mathcal{N}(\mathbf{0},\mathbf{I}_d)}[F(\theta + \sigma\mathbf{g})] \qquad (10)$$



(a) boston, Gaussian (b) cpu, Gaussian

(c) wine, Gaussian (d) parkinson, Gaussian

Figure 5. Normalized Frobenius norm error for the gaussian kernel matrix approximation. We compare the same estimators as for pointwise kernel kernel approximation experiments.

for a smoothing parameter $\sigma > 0$. The gradient of the Gaussian smoothing of F is given by the formula:

$$\nabla F_\sigma(\theta) = \frac{1}{\sigma}\mathbb{E}_{\mathbf{g}\in\mathcal{N}(0,\mathbf{I}_d)}[F(\theta + \sigma\mathbf{g})\mathbf{g}]. \tag{11}$$

The above formula leads to several MC estimators of $\nabla F_\sigma(\theta)$ using as vectors $\mathbf{g}$ the rows of matrices sampled from certain distributions (Conn et al., 2009; Salimans et al., 2017). In particular, it was recently shown that exact OMCs provide in that setting more accurate estimators of $\nabla F_\sigma(\theta)$ that in turn lead to more efficient blackbox optimization algorithms applying gradient-based methods with the estimated gradients used to find maxima/minima of blackbox functions. In the reinforcement learning (RL) setting the blackbox function F takes as input the parameters θ of a policy $\pi_\theta : \mathcal{S} \rightarrow \mathcal{A}$ (mapping states to actions that should be applied in that state), usually encoded by feedforward neural networks, and outputs the total reward obtained by an agent applying that policy π in the given environment. We conduct two sets of RL experiments.

OpenAI Gym tasks: We compare different MC estimators on the task of learning a RL policy for the Swimmer task from OpenAI Gym. The policy is encoded by a neural network with two hidden layers of size 41 each and using Toeplitz matrices. The gradient vector is 253-dimensional and we use $k = 253$ samples for each experiment. We compare different MC estimators, including our new constructions. The results are presented in Fig. 6. GORT stands for the exact OMC (using Gaussian orthogonal directions).

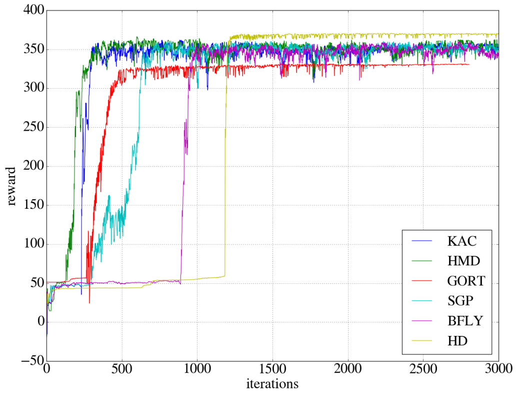


Figure 6. Comparing learning curves for RL policy training for algorithms using different MC estimators to approximate the gradient of the blackbox function on the example of Swimmer task.

Quadruped locomotion with Minitaur platform: We apply Kac's random walk matrices to learn RL walking policies on the simulator of the Minitaur robot. We learn linear policies of 96 parameters. We demonstrate that AOMCs based on Kac's random walk matrices can easily learn good quality walking behaviours (see Appendix for details and full result). We attach a video library showing how these learned walking policies work in practice.

Comments on results: Across Figures 3-5, all OMC/AOMC methods beat IID significantly, confirming earlier observations. Our new HMD approach does particularly well on Frobenius norm, which suggests it may be more effective for downstream tasks. We aim to study this phenomenon in future work. The KAC method performs very well, indeed best in 3 of the 4 datasets in Fig. 3. This is encouraging given our theoretical guarantees in Theorem 5.5, showing KAC works well in practice for small values of the constant C. Another advantage of KAC is that one can use any dimensionality without zero-padding, drastically reducing the number of rollouts required in policy search tasks. In the Swimmer RL task shown in Fig. 6, both HMD and KAC provide excellent performance, rapidly reaching high reward.

7. Conclusion

We have given a unifying account of several approaches for approximately uniform orthogonal matrix generation. Through this unifying perspective, we introduced a new random matrix distribution, Hadamard-MultiRademacher. We also gave the first guarantees that *approximate methods* for OMC can yield statistical improvements relative to baselines, by harnessing recent developments in Kac's random walk theory and conducted extensive empirical evaluation.

Acknowledgements

We thank the anonymous reviewers for helpful comments. MR acknowledges support by EPSRC grant EP/L016516/1 for the Cambridge Centre for Analysis. AW acknowledges support from the David MacKay Newton research fellowship at Darwin College, The Alan Turing Institute under EPSRC grant EP/N510129/1 & TU/B/000074, and the Leverhulme Trust via the CFI.

References

Ailon, N. and Chazelle, B. The fast Johnson-Lindenstrauss transform and approximate nearest neighbors. *SIAM J. Comput.*, 39(1):302–322, 2009.

Andoni, A., Indyk, P., Laarhoven, T., Razenshteyn, I., and Schmidt, L. Practical and optimal LSH for angular distance. In *Neural Information Processing Systems (NIPS)*, 2015.

Choromanski, K. and Sindhwani, V. Recycling randomness with structure for sublinear time kernel expansions. In *International Conference on Machine Learning (ICML)*, 2016.

Choromanski, K., Rowland, M., and Weller, A. The unreasonable effectiveness of structured random orthogonal embeddings. In *Neural Information Processing Systems (NIPS)*, 2017.

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

Choromanski, K., Rowland, M., Sindhwani, V., Turner, R. E., and Weller, A. Structured evolution with compact architectures for scalable policy optimization. In *International Conference on Machine Learning (ICML)*, 2018b.

Conn, A. R., Scheinberg, K., and Vicente, L. N. *Introduction to Derivative-Free Optimization*. SIAM, 2009.

Dasgupta, A., Kumar, R., and Sarlós, T. A sparse Johnson-Lindenstrauss transform. In *Symposium on Theory of Computing (STOC)*, pp. 341–350. ACM, 2010.

Dick, J. and Pillichshammer, F. *Digital Nets and Sequences: Discrepancy Theory and Quasi-Monte Carlo Integration*. Cambridge University Press, 2010.

Genz, A. Methods for generating random orthogonal matrices. In *Monte Carlo and Quasi-Monte Carlo Methods (MCQMC)*, 1998.

Givens, W. Computation of plane unitary rotations transforming a general matrix to triangular form. *Journal of the Society for Industrial and Applied Mathematics*, 6(1): 26–50, 1958.

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

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. A kernel two-sample test. *J. Mach. Learn. Res.*, 13(1):723–773, March 2012.

Jing, L., Shen, Y., Dubcek, T., Peurifoy, J., Skirlo, S. A., LeCun, Y., Tegmark, M., and Soljacic, M. Tunable efficient unitary neural networks (EUNN) and their application to RNNs. In *International Conference on Machine Learning, ICML*, 2017.

Johnson, W. and Lindenstrauss, J. Extensions of Lipschitz mappings into a Hilbert space. In *Conference in Modern Analysis and Probability*, volume 26, pp. 189–206. 1984.

Mathieu, M. and LeCun, Y. Fast approximation of rotations and Hessians matrices. *arXiv*, 2014.

Metropolis, N. and Ulam, S. The Monte Carlo method. *Journal of the American Statistical Association*, 44(247): 335–341, 1949.

Mezzadri, F. How to generate random matrices from the classical compact groups. *Notices of the American Mathematical Society*, 54(5):592 – 604, 5 2007.

Munkhoeva, M., Kapushev, Y., Burnaev, E., and Oseledets, I. Quadrature-based features for kernel approximation. In *Neural Information Processing Systems (NeurIPS)*, 2018.

Nesterov, Y. and Spokoiny, V. Random gradient-free minimization of convex functions. *Found. Comput. Math.*, 17 (2):527–566, April 2017. ISSN 1615-3375.

Oliveira, R. I. On the convergence to equilibrium of Kac's random walk on matrices. *Ann. Appl. Probab.*, 19(3): 1200–1231, 06 2009.

Pillai, N. S. and Smith, A. Kac's walk on n-sphere mixes in $n \log n$ steps. *Ann. Appl. Probab.*, 27(1):631–650, 02 2017.

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

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

Tucker, G., Mnih, A., Maddison, C. J., Lawson, J., and Sohl-Dickstein, J. REBAR: low-variance, unbiased gradient estimates for discrete latent variable models. In *Neural Information Processing Systems (NIPS)*, 2017.

Yu, F., Suresh, A., Choromanski, K., Holtmann-Rice, D., and Kumar, S. Orthogonal random features. In *Neural Information Processing Systems (NIPS)*, 2016.