Random Features for Kernel Approximation: A Survey on Algorithms, Theory, and Beyond

Fanghui Liu *Member, IEEE*, Xiaolin Huang *Senior Member, IEEE*,
Yudong Chen, Johan A.K. Suykens *Fellow, IEEE*

Abstract—The class of random features is one of the most popular techniques to speed up kernel methods in large-scale problems. Related works have been recognized by the NeurIPS Test-of-Time award in 2017 and the ICML Best Paper Finalist in 2019. The body of work on random features has grown rapidly, and hence it is desirable to have a comprehensive overview on this topic explaining the connections among various algorithms and theoretical results. In this survey, we systematically review the work on random features from the past ten years. First, the motivations, characteristics and contributions of representative random features based algorithms are summarized according to their sampling schemes, learning procedures, variance reduction properties and how they exploit training data. Second, we review theoretical results that center around the following key question: how many random features are needed to ensure a high approximation quality or no loss in the empirical/expected risks of the learned estimator. Third, we provide a comprehensive evaluation of popular random features based algorithms on several large-scale benchmark datasets and discuss their approximation quality and prediction performance for classification. Last, we discuss the relationship between random features and modern over-parameterized deep neural networks (DNNs), including the use of high dimensional random features in the analysis of DNNs as well as the gaps between current theoretical and empirical results. This survey may serve as a gentle introduction to this topic, and as a users' guide for practitioners interested in applying the representative algorithms and understanding theoretical results under various technical assumptions. We hope that this survey will facilitate discussion on the open problems in this topic, and more importantly, shed light on future research directions. Due to the page limit, we suggest the readers refer to the full version of this survey https://arxiv.org/abs/2004.11154.

Index Terms—random features, kernel approximation, generalization properties, over-parameterized models

1 Introduction

Kernel methods [1], [2], [3] are one of the most powerful techniques for nonlinear statistical learning problems. Let $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X} \subseteq \mathbb{R}^d$ be two samples and $\phi : \mathcal{X} \to \mathcal{H}$ be a nonlinear feature map transforming each element in $\mathcal{X}$ into a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$, in which the inner product between $\phi(\boldsymbol{x})$ and $\phi(\boldsymbol{x}')$ endowed by $\mathcal{H}$ can be computed using a kernel function $k(\cdot,\cdot) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ as $\langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \rangle_{\mathcal{H}} = k(\boldsymbol{x}, \boldsymbol{x}')$. In practice, the kernel function k is directly given to obtain the inner product $\langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \rangle_{\mathcal{H}}$ instead of finding the explicit expression of ϕ, which is known as the *kernel trick*. Benefiting from this scheme, kernel methods are effective for learning nonlinear structures but often suffer from scalability issues in large-scale problems due to high space and time complexities.

To overcome the poor scalability of kernel methods, the class of random Fourier features (RFFs) [4] is a typical data-independent technique to approximate the kernel function using an explicit feature mapping. RFF applies in particular to shift-invariant (also called "stationary") kernels satisfying $k(\boldsymbol{x}, \boldsymbol{x}') = k(\boldsymbol{x} - \boldsymbol{x}')$. By virtue of the correspondence between a shift-invariant kernel and its Fourier spectral density [5], the kernel can be approximated by $k(\boldsymbol{x}, \boldsymbol{x}') \approx \langle \varphi(\boldsymbol{x}), \varphi(\boldsymbol{x}') \rangle$, where the explicit mapping $\varphi : \mathbb{R}^d \to \mathbb{R}^s$ is obtained by sampling from a distribution defined by the inverse Fourier transform of k. To scale kernel methods in the large sample case (e.g., $n \gg d$), the number of random features s is often taken to be larger than the original sample dimension d but much smaller than the sample size n to achieve computational efficiency in practice[1], e.g., traditional kernel methods [6], [7], neural tangent kernel [8], [9], [10], graph neural networks [11], [12], and attention in Transformers [13], [14]. Interestingly, the random features model can be viewed as a class of two-layer neural networks with fixed weights in the first layer. This connection has important theoretical implications for deep neural networks (DNNs) in the *over-parameterized* regime. Theoretical results [9], [15], [16], [17] for random features can be leveraged to understand DNNs and provide an analysis of two-layer *over-parameterized* neural networks. Partly due to its far-reaching repercussions, the seminal work by Rahimi and Recht on RFF [4] won the Test-of-Time Award in the *Thirty-first Advances in Neural Information Processing Systems* (NeurIPS 2017).

RFF spawns a new direction for kernel approximation, and the past ten years has witnessed a flurry of research papers devoted to this topic. On the algorithmic side, subsequent work has focused on improving the kernel approximation quality [18], [19] and decreasing the time and space complexities [20], [21]. Implementation of RFF has in fact been taken to the hardware level [22], [23]. On the theoretical side, a series of works aim to address the following two key questions:

F. Liu and J.A.K. Suykens are with the Department of Electrical Engineering (ESAT-STADIUS), KU Leuven, B-3001 Leuven, Belgium (email: {fanghui.liu;johan.suykens}@esat.kuleuven.be).
X. Huang is with Institute of Image Processing and Pattern Recognition, and also with Institute of Medical Robotics, Shanghai Jiao Tong University, Shanghai 200240, P.R. China (e-mail: xiaolinhuang@sjtu.edu.cn).
Y. Chen is with School of Operations Research and Information Engineering, Cornell University, Ithaca, NY 14850 USA (e-mail: yudong.chen@cornell.edu).

Manuscript received 04 Aug. 2020; revised 16 March 2021; accepted 06 July 2021. Date of publication xx xxxx 2021; date of current version xx xxxx 2021. (Corresponding author: Fanghui Liu and Xiaolin Huang.)
Recommended for acceptance by xxx.
Digital Object Identifier no. xxx

1. Random features model can be regarded as an over-parameterized model allowing for $s \gg n$, refer to Section 7 for details.

1) **Approximation:** how many random features are needed to ensure high quality of kernel approximation?
2) **Generalization:** how many random features are needed to incur no loss in the expected risk of a learned estimator?

Here "no loss" means how large s should be for the (approximated) kernel estimator with s random features to be almost as good as the exact one. Much research effort has been devoted to this direction, including analyzing the kernel approximation error (the first question above) [4], [24], and studying the risk and generalization properties (the second question above) [7], [25]. Increasingly refined and general results have been obtained over the years. In the *Thirty-sixth International Conference on Machine Learning* (ICML 2019), Li et al. [25] were recognized by the Honorable Mentions (best paper finalist) for their unified theoretical analysis of RFF.

RFF has proved effective in a broad range of machine learning tasks. Given its remarkable empirical success and the rapid growth of the related literature, we believe it is desirable to have a comprehensive overview on this topic summarizing the progress in algorithm design and applications, and elucidating existing theoretical results and their underlying assumptions. With this goal in mind, in this survey we systematically review the work from the past ten years on the algorithms, theory and applications of random features methods. The main contributions of this survey include:

1) We provide an overview of a wide range of random features based algorithms, re-organize the formulation of representative approaches under a unifying framework for a direct understanding and comparison.
2) We summarize existing theoretical results on the kernel approximation error measured in various metrics, as well as results on generalization risk of kernel estimators. The underlying assumptions in these results are discussed in detail. In particular, we (partly) answer an open question in this topic: why good kernel approximation performance cannot lead to good generalization performance?
3) We systematically evaluate and compare the empirical performance of representative random features based algorithms under different experimental settings.
4) We discuss recent research trends on (high dimensional) random features in over-parameterized settings for understanding generalization properties of over-parameterized neural networks as well as the gaps in existing theoretical analysis. We view this topic as a promising research direction.

The remainder of this paper is organized as follows. Section 2 presents the preliminaries and a taxonomy of random features based algorithms. We review *data-independent* algorithms in Section 3 and *data-dependent* approaches in Section 4, respectively. In Section 5, we survey existing theoretical results on kernel approximation and generalization performance. Experimental comparisons of representative random features based methods are given in Section 6. In Section 7, we discuss recent results on random features in over-parameterized regimes. The paper is concluded in Section 8 with a discussion on future directions.

2 Preliminaries and Taxonomies

In this section, we introduce preliminaries on the problem setting and theoretical foundation of random features. We then present a taxonomy of existing random features based algorithms, which sets the stage for the subsequent discussion. A set of commonly used parameters is summarized in Table 1.

Table 1
Commonly used parameters and symbols.

Notation	Definition	Notation	Definition
n	number of samples	d	feature dimension
s	number of random features	λ	regularization parameter
k	(original) kernel function	$\tilde{k}$	(approximated) kernel function
$\boldsymbol{\omega}_i$	random feature	$\boldsymbol{\beta}_\lambda$	optimization variable
$\boldsymbol{x}$	data point	$\boldsymbol{y}$	label vector
ς	Gaussian kernel width	σ	activation function
$\boldsymbol{e}_i$	standard basis vector	u	$u := \langle \boldsymbol{x}, \boldsymbol{x}' \rangle / (\|\boldsymbol{x}\| \|\boldsymbol{x}'\|)$
$\boldsymbol{K}$	(original) kernel matrix	$\widetilde{\boldsymbol{K}}$	(approximated) kernel matrix
$\boldsymbol{\tau}$	$\boldsymbol{\tau} := \boldsymbol{x} - \boldsymbol{x}'$	τ	$\tau := \|\boldsymbol{\tau}\|_2$
$\boldsymbol{Z}$	random feature matrix	$\boldsymbol{W}$	transformation matrix
f_ρ	target function	ℓ	loss function
$f_{\boldsymbol{z},\lambda}$	(original) empirical functional	$\tilde{f}_{\boldsymbol{z},\lambda}$	(approximated) functional
$\mathcal{E}_{\boldsymbol{z}}$	empirical risk	$\mathcal{E}$	expected risk
$l_\lambda(\boldsymbol{\omega})$	ridge leverage function	$d^\lambda_{\boldsymbol{K}}$	effective dimension (matrix)
Σ	integral operator	$\mathcal{N}(\lambda)$	effective dimension (operator)
$\otimes$	tensor product	$\lesssim$	$\leq$ with a constant C times
α	convergence rate for λ	γ	rate for effective dimension

2.1 Problem Settings

Let $\mathcal{X} \subset \mathbb{R}^d$ be a compact metric space of samples, and $\mathcal{Y} = \{-1, 1\}$ (in classification) or $\mathcal{Y} \subseteq \mathbb{R}$ (in regression) be the label space. We assume that a sample set $\{\boldsymbol{z}_i = (\boldsymbol{x}_i, y_i)\}_{i=1}^n$ is drawn from a non-degenerate unknown Borel probability measure ρ on $\mathcal{X} \times \mathcal{Y}$. Let $\mathcal{H}$ be a RKHS endowed with a positive definite kernel function $k(\cdot,\cdot)$, and $\boldsymbol{K} = [k(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{i,j=1}^n$ be the kernel matrix associated with the samples. The *target function* of ρ is defined as $f_\rho(\boldsymbol{x}) = \int_{\mathcal{Y}} y \mathrm{d}\rho(y|\boldsymbol{x})$ for $\boldsymbol{x} \in \mathcal{X}$, where $\rho(\cdot|\boldsymbol{x})$ is the conditional distribution of y given $\boldsymbol{x}$. The typical empirical risk minimization problem is considered as

$$f_{\boldsymbol{z},\lambda} := \operatorname*{argmin}_{f \in \mathcal{H}} \left\{ \frac{1}{n} \sum_{i=1}^n \ell\big(y_i, f(\boldsymbol{x}_i)\big) + \lambda \|f\|_{\mathcal{H}}^2 \right\}, \qquad (1)$$

where $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a loss function and $\lambda \equiv \lambda(n) > 0$ is a regularization parameter. In learning theory, one typically assumes that $\lim_{n\to\infty} \lambda(n) = 0$ and adopts $\lambda := n^{-\alpha}$ with $\alpha \in (0, 1]$.

The loss function $\ell(y, f(\boldsymbol{x}))$ in Eq. (1) measures the quality of the prediction $f(\boldsymbol{x})$ at $\boldsymbol{x} \in \mathcal{X}$ with respect to the observed response y. Popular choices of ℓ include the squared loss $\ell(y, f(\boldsymbol{x})) = (y - f(\boldsymbol{x}))^2$ in kernel ridge regression (KRR) and the hinge loss $\ell(y, f(\boldsymbol{x})) = \max(0, 1 - yf(\boldsymbol{x}))$ in support vector machines (SVMs), etc. For a given ℓ, the empirical risk functional on the sample set is defined as $\mathcal{E}_{\boldsymbol{z}}(f) = \frac{1}{n}\sum_{i=1}^n \ell(y_i, f(\boldsymbol{x}_i))$, and the corresponding expected risk is defined as $\mathcal{E}(f) = \int_{\mathcal{X}\times\mathcal{Y}} \ell(y, f(\boldsymbol{x})) \mathrm{d}\rho$. The statistical theory of supervised learning in an approximation theory view aims to understand the generalization property of $f_{\boldsymbol{z},\lambda}$ as an approximation of the true target function f_ρ, which can be quantified by the excess risk $\mathcal{E}(f_{\boldsymbol{z},\lambda}) - \mathcal{E}(f_\rho)$, or the estimation error $\|f_{\boldsymbol{z},\lambda} - f_\rho\|^2$ in an appropriate norm $\|\cdot\|$.

Using an explicit randomized feature mapping $\varphi : \mathbb{R}^d \to \mathbb{R}^s$, one may approximate the kernel function $k(\boldsymbol{x}, \boldsymbol{x}')$ by $\tilde{k}(\boldsymbol{x}, \boldsymbol{x}') = \langle \varphi(\boldsymbol{x}), \varphi(\boldsymbol{x}') \rangle$. In this case, the approximate kernel $\tilde{k}(\cdot,\cdot)$ defines an RKHS $\widetilde{\mathcal{H}}$ (not necessarily contained in the RKHS $\mathcal{H}$ associated with the original kernel function k). With the above approximation, one solves the following approximate version of problem (1):

$$\tilde{f}_{\boldsymbol{z},\lambda} := \operatorname*{argmin}_{f \in \widetilde{\mathcal{H}}} \left\{ \frac{1}{n} \sum_{i=1}^n \ell\big(y_i, f(\boldsymbol{x}_i)\big) + \lambda \|f\|_{\widetilde{\mathcal{H}}}^2 \right\}. \qquad (2)$$

By the representer theorem [1], the above problem can be rewritten as a finite-dimensional empirical risk minimization problem

$$\boldsymbol{\beta}_\lambda := \underset{\boldsymbol{\beta}\in\mathbb{R}^s}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n} \ell\left(y_i, \boldsymbol{\beta}^\top \varphi(\boldsymbol{x}_i)\right) + \lambda\|\boldsymbol{\beta}\|_2^2 . \tag{3}$$

For example, in least squares regression where ℓ is the squared loss, the first term in problem (3) is equivalent to $\|\boldsymbol{y} - \boldsymbol{Z}\boldsymbol{\beta}\|_2^2$, where $\boldsymbol{y} = [y_1, y_2, \cdots, y_n]^\top$ is the label vector and $\boldsymbol{Z} = [\varphi(\boldsymbol{x}_1), \cdots, \varphi(\boldsymbol{x}_n)]^\top \in \mathbb{R}^{n\times s}$ is the random feature matrix. This is a linear ridge regression problem in the space spanned by the random features, with the optimal prediction given by $\tilde{f}_{\boldsymbol{z},\lambda}(\boldsymbol{x}') = \boldsymbol{\beta}_\lambda^\top \varphi(\boldsymbol{x}')$ for a new data point $\boldsymbol{x}'$, where $\boldsymbol{\beta}_\lambda$ has the explicit expression $\boldsymbol{\beta}_\lambda = (\boldsymbol{Z}^\top\boldsymbol{Z} + n\lambda\boldsymbol{I})^{-1}\boldsymbol{Z}^\top\boldsymbol{y}$.

2.2 Theoretical Foundation of Random Features

The original RFF [4] is used for shift-invariant kernels, which builds on Bochner's theorem [5]: Let $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a bounded, continuous, positive definite, and shift-invariant kernel, it can be represented as the Fourier transform of a finite non-negative Borel measure μ (normalized to be a probability measure $p(\cdot)$ by setting $k(\boldsymbol{0}) = 1$ throughout the paper) on $\mathbb{R}^d$, i.e.

$$\begin{aligned} k(\boldsymbol{x} - \boldsymbol{x}') &= \int_{\mathbb{R}^d} \exp\left(\mathrm{i}\boldsymbol{\omega}^\top(\boldsymbol{x} - \boldsymbol{x}')\right)\mu(\mathrm{d}\boldsymbol{\omega}) \\ &= \mathbb{E}_{\boldsymbol{\omega}\sim p(\cdot)}\left[\exp(\mathrm{i}\boldsymbol{\omega}^\top\boldsymbol{x})\exp(\mathrm{i}\boldsymbol{\omega}^\top\boldsymbol{x}')^*\right], \end{aligned} \tag{4}$$

where the symbol $\boldsymbol{z}^*$ denotes the complex conjugate of $\boldsymbol{z}$. The kernels used in practice are typically real-valued and thus the imaginary part in Eq. (4) can be discarded. According to Eq. (4), RFF makes use of the standard Monte Carlo sampling scheme to approximate $k(\boldsymbol{x}, \boldsymbol{x}')$. In particular, one uses the approximation

$$k(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{E}_{\boldsymbol{\omega}\sim p}[\varphi_p(\boldsymbol{x})^\top\varphi_p(\boldsymbol{x}')] \approx \tilde{k}_p(\boldsymbol{x}, \boldsymbol{x}') := \varphi_p(\boldsymbol{x})^\top\varphi_p(\boldsymbol{x}')$$

with the explicit feature mapping[2]

$$\varphi_p(\boldsymbol{x}) := \frac{1}{\sqrt{s}}\left[\exp(-\mathrm{i}\boldsymbol{\omega}_1^\top\boldsymbol{x}), \cdots, \exp(-\mathrm{i}\boldsymbol{\omega}_s^\top\boldsymbol{x})\right]^\top, \tag{5}$$

where $\{\boldsymbol{\omega}_i\}_{i=1}^s$ are sampled from $p(\cdot)$ *independently* of the training set. Consequently, the original kernel matrix $\boldsymbol{K} = [k(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{n\times n}$ can be approximated by $\boldsymbol{K} \approx \widetilde{\boldsymbol{K}}_p = \boldsymbol{Z}_p\boldsymbol{Z}_p^\top$ with $\boldsymbol{Z}_p = [\varphi_p(\boldsymbol{x}_1), \cdots, \varphi_p(\boldsymbol{x}_n)]^\top \in \mathbb{R}^{n\times s}$. It is convenient to introduce the shorthand $z_p(\boldsymbol{\omega}_i, \boldsymbol{x}_j) := \exp(-\mathrm{i}\boldsymbol{\omega}_i^\top\boldsymbol{x}_j)$ such that $\varphi_p(\boldsymbol{x}) = 1/\sqrt{s}[z_p(\boldsymbol{\omega}_1, \boldsymbol{x}), \cdots, z_p(\boldsymbol{\omega}_s, \boldsymbol{x})]^\top$. With this notation, the approximate kernel $\tilde{k}_p(\boldsymbol{x}, \boldsymbol{x}')$ can be rewritten as $\tilde{k}_p(\boldsymbol{x}, \boldsymbol{x}') = \frac{1}{s}\sum_{i=1}^s z_p(\boldsymbol{\omega}_i, \boldsymbol{x})z_p(\boldsymbol{\omega}_i, \boldsymbol{x}')$.

A similar characterization in Eq. (4) is available for rotation-invariant kernels, where the Fourier basis functions are *spherical harmonics* [26], [27]. Rotation-invariant kernels are dot-product kernels defined on the unit sphere $\mathcal{X} = \mathbb{S}^{d-1} := \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\|_2 = 1\}$. Let $k : \mathbb{S}^d \times \mathbb{S}^d \to \mathbb{R}$ be a bounded, continuous, positive definite, and rotation-invariant kernel, it can be represented as a non-negative expansion with spherical harmonics, i.e.

$$k(\boldsymbol{x}, \boldsymbol{x}') \equiv k(\langle\boldsymbol{x}, \boldsymbol{x}'\rangle) = \sum_{i=0}^{\infty}\Lambda_i \sum_{j=1}^{N(d,i)} Y_{i,j}(\boldsymbol{x})Y_{i,j}(\boldsymbol{x}'),$$

where $\Lambda_i \geq 0$ are the Fourier coefficients, $Y_{i,j}$ is the spherical harmonics, and $N(d,i) = \frac{2i+d-2}{i}\begin{pmatrix} i+d-3 \\ d-2 \end{pmatrix}$, refer to the book [28] for details.

Note that, dot product kernels defined in $\mathbb{R}^d$ do not belong to the *rotation-invariant* class. Nevertheless, by virtue of the neural network structure under Gaussian initialization, some dot product kernels defined on $\mathbb{R}^d$ are able to benefit from the sampling framework behind RFF. Given a two-layer network of the form $f(\boldsymbol{x}; \boldsymbol{\theta}) = \sqrt{\frac{2}{s}}\sum_{j=1}^s a_j\sigma(\boldsymbol{\omega}_j^\top\boldsymbol{x})$ with s neurons (notation chosen to be consistent with the number of random features), for some activation function σ and $\boldsymbol{x} \in \mathbb{R}^d$, when $\boldsymbol{\omega} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$ are fixed and only the second layer (parameters $\boldsymbol{a}$) are optimized[3], this actually corresponds to random features approximation

$$k(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{E}_{\boldsymbol{\omega}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I}_d)}[\sigma(\boldsymbol{\omega}^\top\boldsymbol{x})\sigma(\boldsymbol{\omega}^\top\boldsymbol{x}')], \tag{6}$$

where the nonlinear activation function $\sigma(\cdot)$ depends on the kernel type such that $\varphi(\boldsymbol{x}_i) := \sigma(\boldsymbol{W}\boldsymbol{x}_i)$ in Eq. (5), by denoting the transformation matrix $\boldsymbol{W} := [\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \cdots, \boldsymbol{\omega}_s]^\top \in \mathbb{R}^{s\times d}$. The formulation in (6) is quite general to cover a series of kernels by various activation functions. For example, if we take $\sigma(x) = [\cos(x), \sin(x)]^\top$, Eq. (6) corresponds to the Gaussian kernel, which is the standard RFF model [4] for Gaussian kernel approximation. If we consider the commonly used ReLU activation $\sigma(x) = \max\{0, x\}$ in neural networks, Eq. (6) corresponds to the first order arc-cosine kernel, termed as $k(\boldsymbol{x}, \boldsymbol{x}') \equiv \kappa_1(u) = \frac{1}{\pi}(u(\pi - \arccos(u)) + \sqrt{1-u^2})$ by setting $u := \langle\boldsymbol{x}, \boldsymbol{x}'\rangle/(\|\boldsymbol{x}\|\|\boldsymbol{x}'\|)$. If the Heaviside step function $\sigma(x) = \frac{1}{2}(1 + \operatorname{sign}(x))$ is used, Eq. (6) corresponds to the zeroth order arc-cosine kernel, termed as $k(\boldsymbol{x}, \boldsymbol{x}') \equiv \kappa_0(u) = 1 - \frac{1}{\pi}\arccos(u)$ by setting $u := \langle\boldsymbol{x}, \boldsymbol{x}'\rangle/(\|\boldsymbol{x}\|\|\boldsymbol{x}'\|)$, refer to arc-cosine kernels [30] for details. If we take other activation functions used in neural networks, e.g., erf activations [31], GELU [32] in Eq. (6), such two-layer neural network also corresponds to a kernel. In this case, the standard RFF model is still valid (via Monte Carlo sampling from a Gaussian distribution) for these non-stationary kernels.

Further, for a fully-connected deep neural network (more than two layers) and fixed random weights before the output layer, if the hidden layers are wide enough, one can still approach a kernel obtained by letting the widths tend to infinity [33], [34]. If both intermediate layers and the output layer are trained by (stochastic) gradient descent, for the network $f(\boldsymbol{x}; \boldsymbol{\theta})$ with large enough s, the model remains close to its linearization around its random initialization throughout training, known as *lazy training* regime [35]. Learning is then equivalent to a kernel method with another architecture-specific kernel, known as *neural tangent kernel* (NTK, [8]). Interestingly, NTK for two-layer ReLU networks [36] can be constructed by arc-cosine kernels, i.e., $k(\boldsymbol{x}, \boldsymbol{x}') = \|\boldsymbol{x}\|\|\boldsymbol{x}'\|[u\kappa_0(u) + \kappa_1(u)]$. In fact, there is an interesting line of work showing insightful connections between kernel methods and (over-parameterized) neural networks, but this is out of scope of this survey on random features. We suggest the readers refer to some recent literature [9], [37], [38] for details.

2.3 Used Kernels in Random Features

Most random features based algorithms focus on the Gaussian kernel, which is arguably the most important member of shift-invariant kernels, given by $k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\|\boldsymbol{x} - \boldsymbol{x}'\|_2^2/2\varsigma^2\right)$ with the kernel width $\varsigma > 0$. Its density associated with the Gaussian kernel is Gaussian $\boldsymbol{\omega} \sim \mathcal{N}(0, \varsigma^{-2}\boldsymbol{I}_d)$ as indicated by Bochner's theorem or Eq. (6). Besides, quite a number of random features based approaches focus on another class of kernels

2. The subscript in φ_p, $\boldsymbol{Z}_p$, k_p (and other symbols) emphasizes the dependence on the distribution $p(\cdot)$ but can be omitted for notational simplicity.

3. Extreme learning machine [29] is another structure in a two-layer feedforward neural network by randomly hidden nodes.

admitting Eq. (6) by sampling from the Gaussian distribution $\mathcal{N}(0, \boldsymbol{I}_d)$, e.g., arc-cosine kernels [30], that can be connected to a two-layer neural networks with various activation functions.

Apart from the used Gaussian kernel and arc-cosine kernels, polynomial kernels $k(\boldsymbol{x}, \boldsymbol{x}') = (1 + \langle \boldsymbol{x}, \boldsymbol{x}' \rangle)^b$ with the order b defined in $\mathbb{R}^d$ are a widely used family of dot-product kernels. However, dot-product kernel defined in $\mathbb{R}^d$ admit neither *spherical harmonics* nor Eq. (6). As a result, random features for polynomial kernels work in different theoretical foundations and settings, and have been studied in a smaller number of papers, including Maclaurin expansion [39], the tensor sketch technique [40], [41], and oblivious subspace embedding [42], [43]. Interestingly, if the data are ℓ_2 normalized, dot product kernels defined in $\mathbb{R}^d$ can be transformed as stationary but indefinite (real, symmetric, but not positive definite) on the unit sphere[4]. The related random features based algorithms under this setting provide biased estimators [44], [45], or unbiased estimation [46].

2.4 Taxonomy of random features based algorithms

The key step in algorithm is constructing the mapping

$$\varphi(\boldsymbol{x}) := \frac{1}{\sqrt{s}} [a_1 \exp(-\mathrm{i}\boldsymbol{\omega}_1^\top \boldsymbol{x}), \cdots, a_s \exp(-\mathrm{i}\boldsymbol{\omega}_s^\top \boldsymbol{x})]^\top \quad (7)$$

to approximate the integral (4). Existing algorithms differ in how they select the points $\boldsymbol{\omega}_i$ and weights a_i. Figure 1 presents a taxonomy of some representative random features based algorithms. They can be grouped into two categories, *data-independent* algorithms and *data-dependent* algorithms, based on whether or not the selection of $\boldsymbol{\omega}_i$ and a_i is independent of the training data. Data-independent random features based algorithms can be further categorized into three classes according to their sampling strategy:

i) *Monte Carlo sampling*: The points $\{\boldsymbol{\omega}_i\}_{i=1}^s$ are sampled from $p(\cdot)$ in Eq. (4) (see the red box in Figure 1). In particular, to approximate the Gaussian kernel by RFF [4], these points are sampled from the Gaussian distribution $p = \mathcal{N}(0, \varsigma^{-2}\boldsymbol{I}_d)$, with the weights being equal, i.e., $a_i \equiv 1$ in Eq. (7). To reduce the storage and time complexity, one may replace the dense Gaussian matrix in RFF by structural matrices; see, e.g., Fastfood [47] using Hadamard matrices as well as its general version $\mathcal{P}$-model [48]. An alternative approach is using circulant matrices; see, e.g., Signed Circulant Random Features (SCRF) [49]. To improve the approximation quality, a simple and effective approach is to use an ℓ_2-normalization scheme, which leads to Normalized RFF (NRFF) [50]. Another powerful technique for variance reduction is orthogonalization to decrease the randomness in Monte Carlo sampling. Typical algorithms include Orthogonal Random Features (ORF) [18] by employing an orthogonality constraint to the random Gaussian matrix, Structural ORF (SORF) [18], [72], and Random Orthogonal Embeddings (ROM) [51].

ii) *Quasi-Monte Carlo sampling*: This is a typical sampling scheme in sampling theory [73] to reduce the randomness in Monte Carlo sampling for variance reduction. It can significantly improve the convergence of Monte Carlo sampling by virtue of a low-discrepancy sequence $\boldsymbol{t}_1, \boldsymbol{t}_2, \cdots, \boldsymbol{t}_s \in [0, 1]^d$ instead of a uniform sampling sequence over the unit cube to construct the sample points; see the integral representation in the green box in Figure 1. Based on this representation, it can be used for kernel approximation, as conducted by [19]. Subsequently, Lyu [53] proposes Spherical Structural Features (SSF), which generates asymptotically uniformly distributed points on $\mathbb{S}^{d-1}$ to achieve better convergence rate and approximation quality. The Moment Matching (MM) scheme [54] is based on the same integral representation but uses a d-dimensional refined uniform sampling sequence $\{\boldsymbol{t}_i\}_{i=1}^s$ instead of a low discrepancy sequence. Strictly speaking, SSF and MM go beyond the QMC framework. Nevertheless, these methods share the same integration formulation with QMC over the unit cube and thus we include them here for a streamlined presentation.

iii) *Quadrature based methods*: Numerical integration techniques can be also used to approximate the integral representation in Eq. (4). These techniques may involve *deterministic* selection of the points and weights, e.g., by using Gaussian Quadrature (GQ) [20] or Sparse Grids Quadrature (SGQ) [20] over each dimension (their integration formulation can be found in the first blue box in Figure 1). The selection can also be *randomized*. For example, in the work [21], the d-dimensional integration in Eq. (4) is transformed to a double integral, and then approximated by using the Stochastic Spherical-Radial (SSR) rule (see the second blue box in Figure 1).

Data-dependent algorithms use the training data to guide the selection of points and weights in the random features for better approximation quality and/or generalization performance. These algorithms can be grouped into three classes according to how the random features are generated.

i) Leverage score sampling: Built upon the importance sampling framework, this class of algorithm replaces the original distribution $p(\boldsymbol{\omega})$ by a carefully chosen distribution $q(\boldsymbol{\omega})$ constructed using leverage scores [74], [75] (see the yellow box in Figure 1). The representative approach in this class is Leverage Score based RFF (LS-RFF) [25], and its accelerated version [55], [56].

ii) Re-weighted random feature selection: Here the basic idea is to re-weight the random features by solving a constrained optimization problem. Examples of this approach include weighted RFF [58], [59], weighted QMC [19], and weighted GQ [20]. Note that these algorithms directly learn the weights of pre-given random features. Another line of methods re-weight the random features using a two-step procedure: i) "up-projection": first generate a large set of random features $\{\boldsymbol{\omega}_i\}_{i=1}^J$; ii) "compression": then reduce these features to a small number (e.g., $10^2 \sim 10^3$) in a data-dependent manner, e.g., by using kernel alignment [60], kernel polarization [61], or compressed low-rank approximation [62].

iii) Kernel learning by random features: This class of methods aim to learn the spectral distribution of kernel *from the data* so as to achieve better similarity representation and prediction. Note that these methods learn both the weights and the distribution of the features, and hence differ from the other random features selection methods mentioned above, which assume that the candidate features are generated from a pre-given distribution and only learn the weights of these features. Representative approaches for kernel learning involve a *one-stage* [63] or *two-stage* procedure [64], [65], [66], [67], [68], [69]. From a more general point of view, the aforementioned *re-weighted random features selection* methods can also be classified into this class. Since these methods belong to the broad area of kernel learning instead of kernel approximation, we do not detail them in this survey.

Besides the above three main categories, other data-dependent approaches include the following. i) Quantization random features [70], [76]: under a given memory budget. One interesting observation is that random features achieve better generalization performance than Nyström approximation [77] under the same

4. This setting cannot ensure the data are i.i.d on the unit sphere, which is different from the setting of previously discussed rotation invariant kernels.

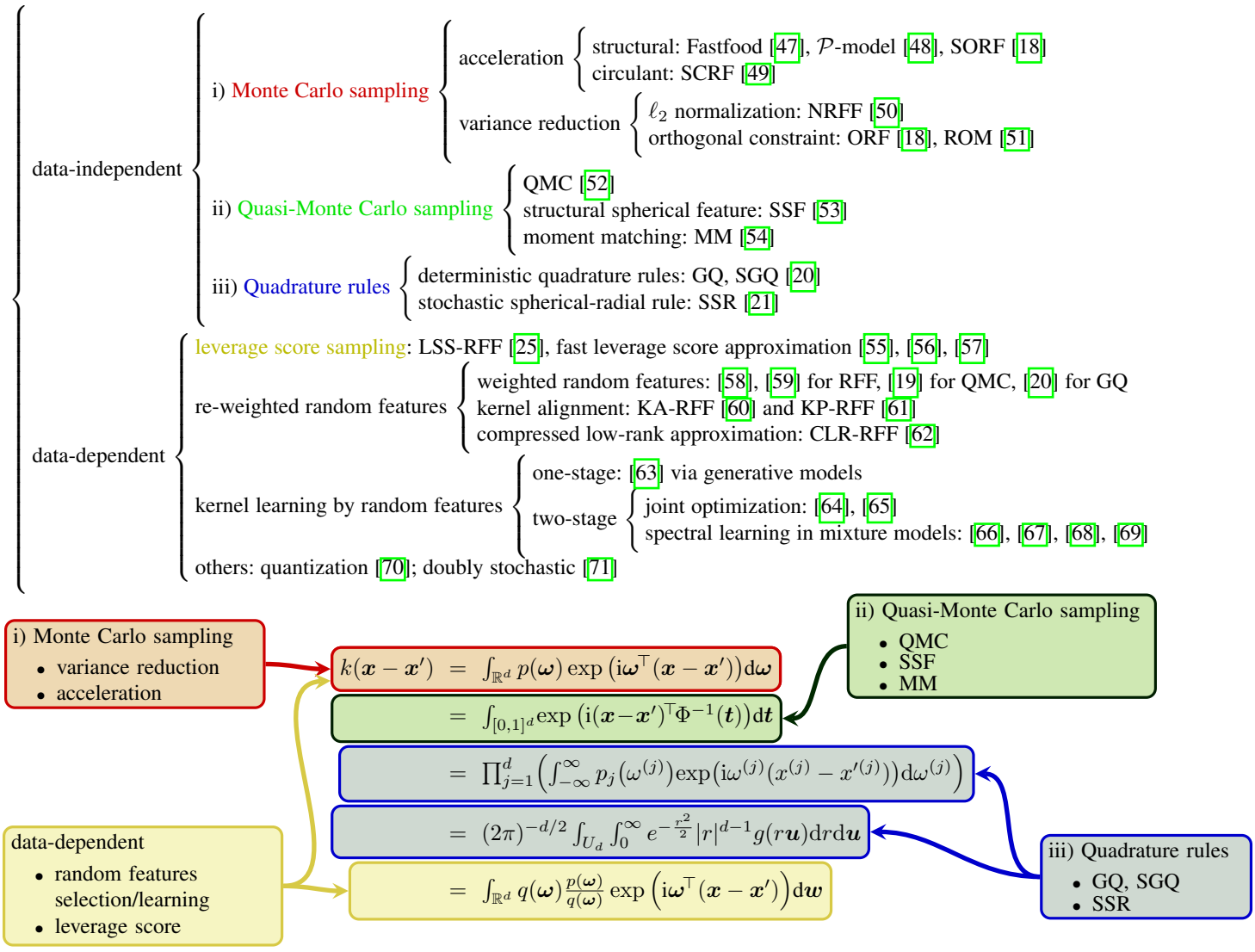


Figure 1. A taxonomy of representative random features based algorithms.

memory space. ii) Doubly stochastic random features [71]: This method uses two sources of stochasticity, one from sampling data points by stochastic gradient descent (SGD), and the other from using RFF to approximate the kernel. This scheme has been used for Kernel PCA approximation [78], and can be further extended to triply stochastic scheme for multiple kernel approximation [79].

3 Data-Independent Algorithms

In this section, we discuss data-independent algorithms in a unified framework based on the transformation matrix $\boldsymbol{W}$, that plays an important role in constructing the mapping $\varphi(\cdot)$ in Eq. (7) and determining how well the estimated kernel converges to the actual kernel. Table 2 reports various random features based algorithms in terms of the class of kernels they apply to (in theory) as well as their space and time complexities for computing the feature mapping $\boldsymbol{W}\boldsymbol{x}$ for a given $\boldsymbol{x} \in \mathcal{X}$. In Table 2, we also summarize the *variance reduction* properties of these algorithms, i.e., whether the variance of the resulting kernel estimator is smaller than the standard RFF. Before proceeding, we introduce some notations and definitions. When discussing a stationary kernel function $k(\boldsymbol{x},\boldsymbol{x}') = k(\boldsymbol{x}-\boldsymbol{x}')$, we use the convenient shorthands $\boldsymbol{\tau} := \boldsymbol{x}-\boldsymbol{x}'$ and $\tau := \|\boldsymbol{\tau}\|_2$. For a random features algorithm $\mathtt{A}$ with frequencies $\{\boldsymbol{\omega}_i\}_{i=1}^s$ sampled from a distribution $p(\cdot)$, we define its expectation $\mathbb{E}(\mathtt{A}) := \mathbb{E}[k(\boldsymbol{\tau})] = \mathbb{E}_{\boldsymbol{\omega}\sim p}\left[1/s\sum_{i=1}^s \cos(\boldsymbol{\omega}_i^\top\boldsymbol{\tau})\right]$ and variance $\mathbb{V}[\mathtt{A}] := \mathbb{V}[k(\boldsymbol{\tau})] = \mathbb{V}\left[\frac{1}{s}\sum_{i=1}^s \cos(\boldsymbol{\omega}^\top\boldsymbol{\tau})\right]$.

3.1 Monte Carlo sampling based approaches

We describe several representative data-independent algorithms based on Monte Carlo sampling, using the Gaussian kernel $k(\boldsymbol{x},\boldsymbol{x}') = k(\boldsymbol{\tau}) = \exp(-\|\boldsymbol{\tau}\|_2^2/2\varsigma^2)$ as an example. Note that these algorithms often apply to more general classes of kernels, as summarized in Table 2.

RFF [4]: For Gaussian kernels, RFF directly samples the random features from a Gaussian distribution (corresponds to the inverse Fourier transform): $\{\boldsymbol{\omega}\}_{i=1}^s \sim p(\boldsymbol{\omega})$. In particular, the corresponding transformation matrix is given by

$$\boldsymbol{W}_{\mathrm{RFF}} = \frac{1}{\varsigma}\boldsymbol{G}, \tag{8}$$

where $\boldsymbol{G} \in \mathbb{R}^{s\times d}$ is a (dense) Gaussian matrix with $G_{ij} \sim \mathcal{N}(0,1)$. For other stationary kernels, the associated $p(\cdot)$ corresponds to the specific distribution given by the Bochner's Theorem. For example, the Laplacian kernel $k(\boldsymbol{\tau}) = \exp(-\|\boldsymbol{\tau}\|_1/\varsigma)$ is

Table 2
Comparison of different kernel approximation methods on space and time complexities to obtain $\boldsymbol{W}\boldsymbol{x}$.

Method	Kernels (in theory)	Extra Memory	Time	Lower variance than RFF
Random Fourier Features (RFF) [4]	shift-invariant kernels	$\mathcal{O}(sd)$	$\mathcal{O}(sd)$	-
Quasi-Monte Carlo (QMC) [52]	shift-invariant kernels	$\mathcal{O}(sd)$	$\mathcal{O}(sd)$	Yes
Normalized RFF (NRFF) [50]	Gaussian kernel	$\mathcal{O}(sd)$	$\mathcal{O}(sd)$	Yes
Moment matching (MM) [54]	shift-invariant kernels	$\mathcal{O}(sd)$	$\mathcal{O}(sd)$	Yes
Orthogonal Random Feature (ORF) [18]	Gaussian kernel	$\mathcal{O}(sd)$	$\mathcal{O}(sd)$	Yes
Fastfood [47]	Gaussian kernel	$\mathcal{O}(s)$	$\mathcal{O}(s\log d)$	No
Spherical Structured Features (SSF) [53]	shift and rotation-invariant kernels	$\mathcal{O}(s)$	$\mathcal{O}(s\log d)$	Yes
Structured ORF (SORF) [18], [72]	shift and rotation-invariant kernels	$\mathcal{O}(s)$	$\mathcal{O}(s\log d)$	Unknown
Signed Circulant (SCRF) [49]	shift-invariant kernels	$\mathcal{O}(s)$	$\mathcal{O}(s\log d)$	The same
$\mathcal{P}$-model [48]	shift and rotation-invariant kernels	$\mathcal{O}(s)$	$\mathcal{O}(s\log d)$	No
Random Orthogonal Embeddings (ROM) [51]	rotation-invariant kernels	$\mathcal{O}(d)$	$\mathcal{O}(d\log d)$	Yes
Gaussian Quadrature (GQ), Sparse Grids Quadrature (SGQ) [20]	shift invariant kernels	$\mathcal{O}(d)$	$\mathcal{O}(d\log d)$	Yes
Stochastic Spherical-Radial rules (SSR) [21]	shift and rotation-invariant kernels	$\mathcal{O}(d)$	$\mathcal{O}(d\log d)$	Yes

associated with a Cauchy distribution. RFF is unbiased, i.e., $\mathbb{E}[\text{RFF}] = \exp(-\|\boldsymbol{\tau}\|_2^2/2\varsigma^2)$, and the corresponding variance is $\mathbb{V}[\text{RFF}] = (1-e^{-\tau^2})^2/2s$.

Fastfood [47]: By observing the similarity between the dense Gaussian matrix and Hadamard matrices with diagonal Gaussian matrices, Le et al. [47] firstly introduce Hadamard and diagonal matrices to speed up the construction of dense Gaussian matrices in RFF, especially in high dimensions (e.g., $d \geq 1000$). In particular, $\boldsymbol{W}$ used in Eq. (8) is substituted by

$$\boldsymbol{W}_{\text{Fastfood}} = \frac{1}{\varsigma}\boldsymbol{B}_1\boldsymbol{H}\boldsymbol{G}\boldsymbol{\Gamma}\boldsymbol{H}\boldsymbol{B}_2\,, \tag{9}$$

where $\boldsymbol{H}$ is the Walsh-Hadamard matrix admitting fast multiplication in $\mathcal{O}(d\log d)$ time, and $\boldsymbol{\Gamma} \in \{0,1\}^{d\times d}$ is a permutation matrix that decorrelates the eigen-systems of two Hadamard matrices. The three *diagonal* random matrices $\boldsymbol{G}$, $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$ are specified as follows: $\boldsymbol{G}$ has independent Gaussian entries drawn from $\mathcal{N}(0,1)$; $\boldsymbol{B}_1$ is a random scaling matrix with $(\boldsymbol{B}_1)_{ii} = \|\boldsymbol{\omega}_i\|_2/\|\boldsymbol{G}\|_{\text{F}}$, which encodes the spectral properties of the associated kernel; $\boldsymbol{B}_2$ is a binary decorrelation matrix with independent random $\{\pm 1\}$ entries. FastFood is an unbiased estimator, but may have a larger variance than RFF: $\mathbb{V}[\text{Fastfood}] \leq \mathbb{V}[\text{RFF}] + \frac{6\tau^4}{s}\left(e^{-\tau^2} + \frac{\tau^2}{3}\right)$, which converges at an $\mathcal{O}(1/s)$ rate.

$\mathcal{P}$-model [48]: A general version of Fastfood, the $\mathcal{P}$-model constructs the transformation matrix as

$$\boldsymbol{W}_{\mathcal{P}} = [\boldsymbol{g}^\top\boldsymbol{P}_1, \boldsymbol{g}^\top\boldsymbol{P}_2, \cdots, \boldsymbol{g}^\top\boldsymbol{P}_s]^\top \in \mathbb{R}^{s\times d}\,,$$

where $\boldsymbol{g}$ is a Gaussian random vector of length a and $\mathcal{P} = \{\boldsymbol{P}_i\}_{i=1}^s$ is a sequence of a-by-d matrices each with unit ℓ_2 norm columns. Fastfood can viewed as a special case of the $\mathcal{P}$-model: the matrix $\boldsymbol{HG}$ in Eq. (9) can be constructed by using a fixed budget of randomness in $\boldsymbol{g}$ and letting each $\boldsymbol{P}_i$ be a random diagonal matrix with diagonal entries of the form $H_{i1}, H_{i2}, \ldots, H_{id}$. The $\mathcal{P}$-model is unbiased and its variance is close to that of RFF with an $\mathcal{O}(1/d)$ convergence rate.

SCRF [49]: It accelerates the construction of random features by using circulant matrices. The transformation matrix is

$$\boldsymbol{W}_{\text{SCRF}} = [\boldsymbol{\nu}\otimes\mathcal{C}(\boldsymbol{\omega}_1), \boldsymbol{\nu}\otimes\mathcal{C}(\boldsymbol{\omega}_2), \cdots, \boldsymbol{\nu}\otimes\mathcal{C}(\boldsymbol{\omega}_t)]^\top \in \mathbb{R}^{td\times d}\,,$$

where $\otimes$ denotes the tensor product, $\boldsymbol{\nu} = [\nu_1, \nu_2, \ldots, \nu_d]$ is a Rademacher vector with $\mathbb{P}(\nu_i = \pm 1) = 1/2$, and $\mathcal{C}(\boldsymbol{w}_i) \in \mathbb{R}^{d\times d}$ is a circulant matrix generated by the vector $\boldsymbol{\omega}_i \sim \mathcal{N}(0, \varsigma^{-2}\boldsymbol{I}_d)$. Thanks to the circulant structure, we only need $\mathcal{O}(s)$ space to store the feature mapping matrix $\boldsymbol{W}_{\text{SCRF}}$ with $s = td$. Note that $\mathcal{C}(\boldsymbol{w}_i)$ can be diagonalized using the Discrete Fourier Transform for $\boldsymbol{\omega}_i$. SCRF is unbiased and has the same variance as RFF.

The above three approaches are designed to accelerate the computation of RFF. We next overview representative methods that aim for better approximation performance than RFF.

NRFF [50]: It normalizes the input data to have unit ℓ_2 norm before constructing the random Fourier features. With normalized data, the Gaussian kernel can be computed as

$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{1}{\varsigma^2}\left(1 - \frac{\boldsymbol{x}^\top\boldsymbol{x}'}{\|\boldsymbol{x}\|_2\|\boldsymbol{x}'\|_2}\right)\right)\,,$$

which is related to the normalized linear kernel [44], [50]. Albeit simple, NRFF is effective in variance reduction and satisfies $\mathbb{V}[\text{NRFF}] = \mathbb{V}[\text{RFF}] - \frac{1}{4s}e^{-\tau^2}(3 - e^{-2\tau^2})$.

ORF [18]: It imposes orthogonality on random features for the Gaussian kernel and has the transformation matrix

$$\boldsymbol{W}_{\text{ORF}} = \frac{1}{\varsigma}\boldsymbol{S}\boldsymbol{Q}\,,$$

where $\boldsymbol{Q}$ is a uniformly distributed random orthogonal matrix, and $\boldsymbol{S}$ is a diagonal matrix with diagonal entries sampled *i.i.d* from the χ-distribution with d degrees of freedom. This orthogonality constraint is useful in reducing the approximation error in random features. It is also considered in [80] for unifying orthogonal Monte Carlo methods. ORF is unbiased and with the variance reduction property $\text{Var}[\text{ORF}] < \text{Var}[\text{RFF}]$ under some conditions, e.g., when d is large and τ is small. For a large d, the ratio of the variances of ORF and RFF can be approximated by $\frac{\mathbb{V}[\text{ORF}]}{\mathbb{V}[\text{RFF}]} \approx 1 - \frac{(s-1)e^{-\tau^2}\tau^4}{d\left(1-e^{-\tau^2}\right)^2}$. Choromanski et al. [81] further improve the variance bound to $\mathbb{V}[\text{ORF}] < \mathbb{V}[\text{RFF}]$, which holds asymptotically in two cases.

SORF [18], [72]: It replaces the random orthogonal matrices used in ORF by a class of structured matrices akin to those in Fastfood. The transformation matrix of SORF is given by

$$\boldsymbol{W}_{\text{SORF}} = \frac{\sqrt{d}}{\varsigma}\boldsymbol{H}\boldsymbol{D}_1\boldsymbol{H}\boldsymbol{D}_2\boldsymbol{H}\boldsymbol{D}_3\,, \tag{10}$$

where $\boldsymbol{H}$ is the normalized Walsh-Hadamard matrix and $\boldsymbol{D}_i \in \mathbb{R}^{d\times d}$, $i = 1, 2, 3$ are diagonal "sign-flipping" matrices, of

which each diagonal entry is sampled from the Rademacher distribution. Bojarski et al. [72] consider more general structures for the three blocks of matrices $\boldsymbol{HD}_i$ in Eq. (10). Note that each block plays a different role. The first block $\boldsymbol{HD}_1$ satisfies $\Pr\left[\|\boldsymbol{HD}_1\boldsymbol{x}\|_\infty > \frac{\log d}{\sqrt{d}}\right] \leq 2de^{-\frac{\log^2 d}{8}}$ for any $\boldsymbol{x} \in \mathbb{R}^d$ with $\|\boldsymbol{x}\|_2 = 1$, termed as $(\log d, 2de^{-\frac{\log^2 d}{8}})$-balanced, hence no dimension carries too much of the ℓ_2 norm of the vector $\boldsymbol{x}$. The second block $\boldsymbol{HD}_2$ ensures that vectors are close to orthogonal. The third block $\boldsymbol{HD}_3$ controls the capacity of the entire structured transform by providing a vector of parameters. SORF is not an unbiased estimator of the Gaussian kernel, but it satisfies an asymptotic unbiased property with $\left|\mathbb{E}\left[\text{SORF}\right] - e^{-\tau^2/2}\right| \leq \frac{6\tau}{\sqrt{d}}$.

ROM [51]: It generalizes SORF to the form

$$\boldsymbol{W}_{\text{ROM}} = \frac{\sqrt{d}}{\varsigma}\prod_{i=1}^{t}\boldsymbol{HD}_i\,,$$

where $\boldsymbol{H}$ can be the normalized Hadamard matrix or the Walsh matrix, and $\boldsymbol{D}_i$ is the Rademacher matrix as defined in SORF. Theoretical results in [51] show that the ROM estimator achieves variance reduction compared to RFF. Interestingly, odd values of t yield better results than even t. This provides an explanation for why SORF chooses $t = 3$.

From the above description, one can find that orthogonalization is a typical operation for variance reduction, e.g., ORF/SORF/ROM. Here we take the Gaussian kernel as an example to illustrate insights of such scheme. By sampling $\{\boldsymbol{\omega}_i\}_{i=1}^s \sim \mathcal{N}(\boldsymbol{0}, \varsigma^{-2}\boldsymbol{I}_d)$, the used Gaussian distribution is isotropic and only depends on the norm $\|\boldsymbol{\omega}\|_2$ instead of $\boldsymbol{\omega}$. The used orthogonal operator makes the direction of $\boldsymbol{\omega}_i$ orthogonal to each other (that means more uniform) while retaining its norm unchanged[5], which leads to decrease the randomness in Monte Carlo sampling, and thus achieve variance reduction effect. If we attempt to directly decrease the randomness in Monte Carlo sampling, QMC is a powerful way to achieve this goal and can then be used to kernel approximation. This is another line of random features with variance reduction illustrated as below.

3.2 Quasi-Monte Carlo Sampling

Here we briefly review methods based on quasi-Monte Carlo sampling (QMC) [52], spherical structured feature (SSF) [53], and moment matching (MM) [54]. These three methods achieve a lower variance or approximation error than RFF.

QMC [52]: It assumes that $p(\cdot)$ factorizes with respect to the dimensions, i.e., $p(\boldsymbol{x}) = \prod_{j=1}^d p_j\left(x_j\right)$, where each $p_j(\cdot)$ is a univariate density function. QMC generally transforms an integral on $\mathbb{R}^d$ in Eq. (4) to one on the unit cube $[0,1]^d$ as

$$k(\boldsymbol{x}-\boldsymbol{x}') = \int_{[0,1]^d}\exp\left(\mathrm{i}(\boldsymbol{x}-\boldsymbol{x}')^\top\Phi^{-1}(\boldsymbol{t})\right)\mathrm{d}\boldsymbol{t}\,, \tag{11}$$

where $\Phi^{-1}(\boldsymbol{t}) = \left(\Phi_1^{-1}\left(t_1\right), \cdots, \Phi_d^{-1}\left(t_d\right)\right) \in \mathbb{R}^d$ with Φ_j being the cumulative distribution function (CDF) of p_j. Accordingly, by generating a low *discrepancy* sequence $\boldsymbol{t}_1, \boldsymbol{t}_2, \cdots, \boldsymbol{t}_s \in [0,1]^d$, the random frequencies can be constructed by $\boldsymbol{\omega}_i = \Phi^{-1}(\boldsymbol{t}_i)$. The corresponding transformation matrix for QMC is

$$\boldsymbol{W}_{\text{QMC}} = [\Phi^{-1}(\boldsymbol{t}_1), \Phi^{-1}(\boldsymbol{t}_2), \cdots, \Phi^{-1}(\boldsymbol{t}_s)]^\top \in \mathbb{R}^{s\times d}\,. \tag{12}$$

QMC achieves an asymptotic error convergence rate of $\mathcal{O}((\log s)^d/s)$, which is faster than the $\mathcal{O}(s^{-1/2})$ rate of MC.

5. In fact, while orthogonalization only makes the direction of $\{\boldsymbol{\omega}_i\}_{i=1}^s$ more uniform, one can make the length $\|\boldsymbol{\omega}_i\|_2$ uniform by sampling from the cumulative distribution function of $\|\boldsymbol{\omega}\|_2$.

SSF [53]: It improves the space and time complexities of QMC for approximating shift- and rotation-invariant kernels. SSF generates points $\{\boldsymbol{v}_i\}_{i=1}^s$ asymptotically uniformly distributed on the sphere $\mathbb{S}^{d-1}$, and construct the transformation matrix as

$$\boldsymbol{W}_{\text{SSF}} = [\Phi^{-1}(t)\boldsymbol{v}_1, \Phi^{-1}(t)\boldsymbol{v}_2, \cdots, \Phi^{-1}(t)\boldsymbol{v}_s]^\top \in \mathbb{R}^{s\times d}\,,$$

where $\Phi^{-1}(t)$ uses the one-dimensional QMC point. The structure matrix $\boldsymbol{V} := [\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_s] \in \mathbb{S}^{(d-1)\times s}$ is a subset of the discrete Fourier matrix by minimizing the discrete Riesz 0-energy [82] such that the points spread as evenly as possible on the sphere.

MM [54]: It also uses the transformation matrix in Eq. (12), but generates a d-dimensional uniform sampling sequence $\{\boldsymbol{t}_i\}_{i=1}^s$ by a moment matching scheme instead of using a low discrepancy sequence as in QMC. In particular, the transformation matrix is

$$\boldsymbol{W}_{\text{MM}} = [\widetilde{\Phi}^{-1}(\boldsymbol{t}_1), \widetilde{\Phi}^{-1}(\boldsymbol{t}_2), \cdots, \widetilde{\Phi}^{-1}(\boldsymbol{t}_s)]^\top \in \mathbb{R}^{s\times d}\,, \tag{13}$$

where one uses moment matching to construct the vectors $\widetilde{\Phi}^{-1}(\boldsymbol{t}_i) = \tilde{\boldsymbol{A}}^{-1}(\Phi^{-1}(\boldsymbol{t}_i) - \tilde{\boldsymbol{\mu}})$ with the sample mean $\tilde{\boldsymbol{\mu}} = \frac{1}{s}\sum_{i=1}^s \Phi^{-1}(\boldsymbol{t}_i)$ and the square root of the sample covariance matrix $\tilde{\boldsymbol{A}}$ satisfying $\tilde{\boldsymbol{A}}\tilde{\boldsymbol{A}}^\top = \text{Cov}(\Phi^{-1}(\boldsymbol{t}_i) - \tilde{\boldsymbol{\mu}})$.

To achieve the target of variance reduction, both orthogonalization in Monte Carlo sampling and QMC based algorithms share the similar principle, namely, generating random features that are as independent/uniform as possible. To be specific, QMC and MM are able to generate more uniform data points to avoid undesirable *clustering* effect, see Figure 1 in [52]. Likewise, SSF aims to generate asymptotically uniformly distributed points on the sphere $\mathbb{S}^{d-1}$, which attempts to encode more information with fewer random features, and thus allows for variance reduction. In sampling theory, QMC can be further improved by an sub-grouped based rank-one lattice construction [83] for computational efficiency, which can be used for the subsequent kernel approximation.

3.3 Quadrature based Methods

Quadrature based methods build on a long line of work on numerical quadrature for estimating integrals. In quadrature methods, the weights are often non-uniform, and the points are usually selected using *deterministic* rules. Below we briefly review these methods.

GQ [20]: It also assumes that the kernel function k factorizes with respect to the dimensions, and thus can be approximated by a one-dimensional Gaussian quadrature rule [84]. For a third-point rule with the points $\{-\hat{p}_1, 0, \hat{p}_1\}$ and their associated weights $(\hat{a}_1, \hat{a}_0, \hat{a}_1)$, the transformation matrix $\boldsymbol{W}_{\text{GQ}} \in \mathbb{R}^{s\times d}$ has entries W_{ij} admitting $\Pr\left(W_{ij} = \pm\hat{p}_1\right) = \hat{a}_1$, $\Pr\left(W_{ij} = 0\right) = \hat{a}_0$. However the total number of the needed points s scales exponentially with the dimension d and thus this method suffers from the curse of dimensionality. To alleviate this, SGQ [20] uses the Smolyak rule [87] to decrease the needed number of points. Here we consider the third-degree SGQ using the symmetric univariate quadrature points $\{-\hat{p}_1, 0, \hat{p}_1\}$ with weights $(\hat{a}_1, \hat{a}_0, \hat{a}_1)$. The corresponding transformation matrix is

$$\boldsymbol{W}_{\text{SGQ}} = [\boldsymbol{0}_d, \hat{p}_1\boldsymbol{e}_1, \cdots, \hat{p}_1\boldsymbol{e}_d, -\hat{p}_1\boldsymbol{e}_1, \cdots, -\hat{p}_1\boldsymbol{e}_d]^\top \in \mathbb{R}^{(2d+1)\times d},$$

where $\boldsymbol{e}_i$ is the d-dimensional standard basis vector with the i-th element being 1.

SSR [21]: It transforms Eq. (6) (actually a d-dimensional integral) to a double integral over a hyper-sphere and the real line. Let $\boldsymbol{\omega} = r\boldsymbol{u}$ with $\boldsymbol{u}^\top\boldsymbol{u} = 1$ for $r \in [0, \infty)$, we have

$$k(\boldsymbol{x}-\boldsymbol{x}') = \frac{C_d}{2}\int_{\mathbb{S}^{d-1}}\int_{-\infty}^{\infty} e^{-\frac{r^2}{2}}|r|^{d-1}g(r\boldsymbol{u})\mathrm{d}r\mathrm{d}\boldsymbol{u}\,, \tag{14}$$

where the integrand is $g(\boldsymbol{\omega}) := \sigma(\boldsymbol{\omega}^\top \boldsymbol{x})\sigma(\boldsymbol{\omega}^\top \boldsymbol{x}')$ given in Eq. (6) and $C_d := (2\pi)^{-d/2}$. The inner integral in Eq. (14) can be approximated by stochastic *radial* rules of degree $2l+1$, i.e., $R(g) = \sum_{i=0}^{l} \hat{w}_i \frac{g(\rho_i)+g(-\rho_i)}{2}$. The outer integral over the d-sphere in Eq. (14) can be approximated by stochastic *spherical* rules: $S_{\mathrm{Q}}(g) = \sum_{j=1}^{q} \widetilde{w}_j g(\boldsymbol{Q}\boldsymbol{u}_j)$, where $\boldsymbol{Q}$ is a random orthogonal matrix and $\widetilde{w}_j$ are stochastic weights whose distributions are such that the rule is exact for polynomials of degree q and gives unbiased estimate for other functions. Combining the above two rules, we have the SSR rule. Accordingly, the transformation matrix of SSR is

$$\boldsymbol{W}_{\mathrm{SSR}} = \boldsymbol{\vartheta} \otimes \begin{bmatrix} (\boldsymbol{Q}\boldsymbol{V})^\top \\ -(\boldsymbol{Q}\boldsymbol{V})^\top \end{bmatrix} \in \mathbb{R}^{2(d+1)\times d},$$

with $\boldsymbol{\vartheta} = [\vartheta_1, \vartheta_2, \cdots, \vartheta_s]$ and $\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_{d+1}]$, where $\vartheta \sim \chi(d+2)$ and $\{\boldsymbol{v}_i\}_{i=1}^{d+1}$ are the vertices of a unit regular d-simplex, which is randomly rotated by $\boldsymbol{Q}$.

In general, according to Eq. (6), kernel approximation by random features is actually a d-dimensional integration approximation problem in mathematics. Sampling methods and quadrature based rules are two typical classes of approaches for high-dimensional integration approximation. Efforts on quadrature based methods focus on developing a high-accuracy, mesh-free, efficiency rule, e.g., [88], [89].

4 Data-dependent algorithms

Data-dependent approaches aim to design/learn the random features using the training data so as to achieve better approximation quality or generalization performance. Based on how the random features are generated, we can group these algorithms into three classes: *leverage score sampling*, *random features selection*, and *kernel learning by random features*. Here we only review leverage score sampling based algorithms due to the page limit.

Leverage score based approaches [25], [56], [92] are built on the *importance sampling* framework. Here one samples $\{\boldsymbol{w}_i\}_{i=1}^{s}$ from a distribution $q(\boldsymbol{w})$ that needs to be designed, and then uses the following feature mapping in Eq. (5):

$$\varphi_q(\boldsymbol{x}) = \frac{1}{\sqrt{s}} \left(\sqrt{\frac{p(\boldsymbol{w}_1)}{q(\boldsymbol{w}_1)}} e^{-\mathrm{i}\boldsymbol{w}_1^\top \boldsymbol{x}}, \cdots, \sqrt{\frac{p(\boldsymbol{w}_s)}{q(\boldsymbol{w}_s)}} e^{-\mathrm{i}\boldsymbol{w}_s^\top \boldsymbol{x}} \right)^\top. \tag{15}$$

Consequently, we have the approximation $k(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{E}_{\boldsymbol{w}\sim q}[\varphi_q(\boldsymbol{x})^\top \varphi_q(\boldsymbol{x}')] \approx \sum_{i=1}^{s} z_q(\boldsymbol{w}_i, \boldsymbol{x}) z_q(\boldsymbol{w}_i, \boldsymbol{x}')$, where $z_q(\boldsymbol{w}_i, \boldsymbol{x}_j) := \sqrt{p(\boldsymbol{w}_i)/q(\boldsymbol{w}_i)} z_p(\boldsymbol{w}_i, \boldsymbol{x}_j)$. Thus, the kernel matrix $\boldsymbol{K}$ can be approximated by $\boldsymbol{K}_q = \boldsymbol{Z}_q \boldsymbol{Z}_q^\top$, where $\boldsymbol{Z}_q := [\varphi_q(\boldsymbol{x}_1), \cdots, \varphi_q(\boldsymbol{x}_n)]^\top \in \mathbb{R}^{n\times s}$. Denoting by $\boldsymbol{z}_{q,\boldsymbol{w}_i}(\boldsymbol{X})$ the i-th column of $\boldsymbol{Z}_q$, we have $\boldsymbol{K} = \mathbb{E}_{\boldsymbol{w}\sim p}[\boldsymbol{z}_{p,\boldsymbol{w}}(\boldsymbol{X}) \boldsymbol{z}_{p,\boldsymbol{w}}^\top(\boldsymbol{X})] = \mathbb{E}_{\boldsymbol{w}\sim q}[\boldsymbol{z}_{q,\boldsymbol{w}}(\boldsymbol{X}) \boldsymbol{z}_{q,\boldsymbol{w}}^\top(\boldsymbol{X})]$.

To design the distribution q, one makes use of the ridge leverage function [74], [75] in KRR:

$$l_\lambda(\boldsymbol{\omega}_i) = p(\boldsymbol{\omega}_i) \boldsymbol{z}_{p,\boldsymbol{\omega}_i}^\top(\boldsymbol{X}) (\boldsymbol{K} + n\lambda \boldsymbol{I})^{-1} \boldsymbol{z}_{p,\boldsymbol{\omega}_i}(\boldsymbol{X}), \tag{16}$$

where λ is the KRR regularization parameter. Define

$$d_{\boldsymbol{K}}^{\lambda} := \int_{\mathbb{R}^d} l_\lambda(\boldsymbol{\omega}) \mathrm{d}\boldsymbol{\omega} = \mathrm{tr}\left[\boldsymbol{K}(\boldsymbol{K} + n\lambda \boldsymbol{I})^{-1}\right]. \tag{17}$$

The quantity $d_{\boldsymbol{K}}^{\lambda} \ll n$ determines the number of independent parameters in a learning problem and hence is referred to as the *number of effective degrees of freedom* [93], [94]. With the above notation, the distribution q designed in [75] is given by

$$q(\boldsymbol{\omega}) := \frac{l_\lambda(\boldsymbol{\omega})}{\int l_\lambda(\boldsymbol{\omega}) \mathrm{d}\boldsymbol{\omega}} = \frac{l_\lambda(\boldsymbol{\omega})}{d_{\boldsymbol{K}}^{\lambda}}. \tag{18}$$

- approximation error
  - $\|k - \tilde{k}\|_\infty$: [4], [24], [98], [99]
  - $\|k - \tilde{k}\|_{L^r}$: [24], [98]
  - Δ-spectral approximation: [75], [81]
  - (Δ_1, Δ_2)-spectral approximation: [70]
- empirical risk: [70], [75]
- expected risk
  - squared loss
    - $\boldsymbol{\omega} \sim p(\boldsymbol{\omega})$: [25], [100]
    - $\boldsymbol{\omega} \sim q(\boldsymbol{\omega})$: [25]
  - Lipschitz continuous
    - $\boldsymbol{\omega} \sim p(\boldsymbol{\omega})$: [25], [101]
    - $\boldsymbol{\omega} \sim q(\boldsymbol{\omega})$: [7], [25], [74]

Figure 2. Taxonomy of theoretical results on random features.

Compared to standard Monte Carlo sampling for RFF, leverage score sampling requires fewer Fourier features and enjoys nice theoretical guarantees [25], [75] (see the next section for details). Note that $q(\boldsymbol{\omega})$ can be also defined by the integral operator [74], [95] rather than the Gram matrix used above, but we do not strictly distinguish these two cases. The typical leverage score based sampling algorithm for RFF is illustrated in [25] as below.

LS-RFF (Leverage Score-RFF) [25]: It uses a subset of data to approximate the matrix $\boldsymbol{K}$ in Eq. (17) so as to compute $d_{\boldsymbol{K}}^{\lambda}$. LS-RFF needs $\mathcal{O}(ns^2 + s^3)$ time to generate refined random features, which can be used in KRR [25] and SVM [7] for prediction.

Note that leverage scores sampling is a powerful tool used in sub-sampling algorithms for approximating large kernel matrices with theoretical guarantees, in particular in Nyström approximation. Research on this topic mainly focuses on obtaining fast leverage score approximation due to inversion of an n-by-n kernel matrix, e.g., two-pass sampling [96] (LS-RFF belongs to this), online setting [97], path-following algorithm [55], or developing various surrogate leverage score sampling based algorithms [56], [57], [92].

5 Theoretical Analysis

In this section, we review a range of theoretical results that center around the two questions mentioned in the introduction. Figure 2 provides a taxonomy of representative work on these two questions.

5.1 Approximation error

Table 3 summarizes representative theoretical results on the convergence rates, the upper bound of the growing diameter, and the resulting sample complexity under different metrics. Here sample complexity means the number of random features sufficient for achieving a maximum approximation error at most ϵ.

The first result of this kind is given by Rahimi and Recht [4], who use a covering number argument to derive a uniform convergence guarantee as follows. For a compact subset $\mathcal{S}$ of $\mathbb{R}^d$, let $|\mathcal{S}| := \sup_{\boldsymbol{x},\boldsymbol{x}'\in\mathcal{S}} \|\boldsymbol{x} - \boldsymbol{x}'\|_2$ be its diameter and consider the L^∞ error $\|k - \tilde{k}\|_\infty := \sup_{\boldsymbol{x},\boldsymbol{x}'\in\mathcal{S}} |k(\boldsymbol{x}, \boldsymbol{x}') - \tilde{k}(\boldsymbol{x}, \boldsymbol{x}')|$.

Theorem 1. *[Uniform convergence of RFF [4], [24]] Let $\mathcal{S}$ be a compact subset of $\mathbb{R}^d$ with diameter $|\mathcal{S}|$. Then, for a stationary kernel k and its approximated kernel $\tilde{k}$ obtained by RFF, we have*

$$\Pr\left[\|k - \tilde{k}\|_\infty \geq \epsilon\right] \leq C_d \left(\frac{\varsigma_p |\mathcal{S}|}{\epsilon}\right)^{\frac{2d}{d+2}} \exp\left(-\frac{s\epsilon^2}{4(d+2)}\right),$$

where $\varsigma_p^2 = \mathbb{E}_p[\boldsymbol{\omega}^\top \boldsymbol{\omega}] = \mathrm{tr}\,\nabla^2 k(0) \in \mathcal{O}(d)$, and $C_d := 2^{\frac{6d+2}{d+2}} \left(\left(\frac{2}{d}\right)^{\frac{d}{d+2}} + \left(\frac{d}{2}\right)^{\frac{2}{d+2}} \right)$ satisfies $C_d \leq 256$ in [4] and is

further improved to $C_d \leq 66$ in [24] by optimization balls of radius in covering number.

According to the above theorem by covering number, with $s := \Omega(\epsilon^{-2} d \log(1/\epsilon\delta))$ random features, one can ensure an ϵ uniform approximation error with probability greater than $1-\delta$. This result also applies to dot-product kernels by random Maclaurin feature maps (see [39, Theorem 8]). The quadrature based algorithm [21] follows this proof framework, and achieves the same error bound with a smaller constant than RFF in Theorem 1 by an extra boundedness assumption. Instead, Fastfood [47] on Gaussian kernels achieves $\mathcal{O}(\sqrt{\log(d/\delta)})$ times approximation error than RFF due to estimates for $\boldsymbol{\Gamma H B}_2$ in Eq. (9), which is based on concentration inequalities for Lipschitz continuous functions under the Gaussian distribution.

Different from the above results using Hoeffding's inequality for the covering number bound in their proof, Sriperumbudur and Szabó [98] revisit the above bound by refined technique of McDiarmid's inequality, symmetrization and bound the expectation of Rademacher average by Dudley entropy bound. Then they provide improved rates (with better constants) from linear dependence on $|\mathcal{S}|$ in Theorem 1 to logarithmic dependence. Apart from the L^∞ error bound, the authors of [98] further derive bounds on the L^r error $\|k-\tilde{k}\|_{L^r} := \left(\int_{\mathcal{S}}\int_{\mathcal{S}} |k(\boldsymbol{x},\boldsymbol{x}') - \tilde{k}(\boldsymbol{x},\boldsymbol{x}')|^r \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{x}'\right)^{1/r}$ for $1 \leq r < \infty$; see Table 3 for a summary. We remark that the $L^2_{\rho_\mathcal{X}}$ error bound is also given in [102], though the rate in [98] is sometimes better in terms of the diameter.

Avron et al. [75] argue that the above point-wise distances $\|k-\tilde{k}\|_\infty$ or $\|k-\tilde{k}\|_{L^r}$ are not sufficient to accurately measure the approximation quality. Instead, they focus on the following spectral approximation criterion.

Definition 1. *[Δ-spectral approximation [75]] For $0 \leq \Delta < 1$, a symmetric matrix $\boldsymbol{A}$ is a Δ-spectral approximation of another symmetric matrix $\boldsymbol{B}$, if $(1-\Delta)\boldsymbol{B} \preceq \boldsymbol{A} \preceq (1+\Delta)\boldsymbol{B}$, where $A \preceq B$ indicates that $B-A$ is a semi-positive definite matrix.*

According to this definition, $\boldsymbol{Z}\boldsymbol{Z}^\top + n\lambda \boldsymbol{I}_n$ is Δ-spectral approximation of $\boldsymbol{K} + n\lambda \boldsymbol{I}_n$ if $(1-\Delta)(\boldsymbol{K}+n\lambda\boldsymbol{I}_n) \preceq \boldsymbol{Z}\boldsymbol{Z}^\top + n\lambda\boldsymbol{I}_n \preceq (1+\Delta)(\boldsymbol{K}+n\lambda\boldsymbol{I}_n)$. Avron et al. [75, Theorem 7] state that $\Omega(n_\lambda \log d^\lambda_{\boldsymbol{K}})$ random features are sufficient to guarantee Δ-spectral approximation by the matrix Bernstein concentration inequality and effective degree of freedom, where $n_\lambda := n/\lambda$. Under this framework, Choromanski et al. [81, Theorem 5.4] present a non-asymptotic comparison result between RFF and ORF for spectral approximation by virtue of the smallest singular value of $\boldsymbol{K}+n\lambda\boldsymbol{I}$. If we consider data-dependent sampling, i.e., $\{\boldsymbol{\omega}_i\}_{i=1}^s \sim q(\cdot)$ in Eq. (18) instead of the standard $p(\boldsymbol{\omega})$, $\Omega(d^\lambda_{\boldsymbol{K}} \log d^\lambda_{\boldsymbol{K}})$ random features are needed to suffice for spectral approximation of $\boldsymbol{K}$, which is less than $\Omega(n_\lambda \log d^\lambda_{\boldsymbol{K}})$ [103].

The authors of [70] generalize the notion of Δ-spectral approximation in Definition 1 to (Δ_1,Δ_2)-spectral approximation such that $(1-\Delta_1)\boldsymbol{B} \preceq \boldsymbol{A} \preceq (1+\Delta_2)\boldsymbol{B}$. This refined definition is motivated by the argument that the quantities Δ_1 and Δ_2 in the upper and lower bounds may have different impact on the generalization performance. Using this definition, Zhang et al. [70] derive the approximation guarantees when one quantizes each random Fourier feature $\boldsymbol{\omega}_i$ to a low-precision b-bit representation, which allows more features to be stored in the same amount of space; see Table 3 for a summary.

5.2 Risk and generalization property

The above results on approximation error are a means to an end. More directly related to the learning performance is understanding generalization properties of random features based algorithms. To this end, a series of work study the generalization properties of algorithms based on $p(\boldsymbol{\omega})$-sampling and $q(\boldsymbol{\omega})$-sampling. Under different assumptions, theoretical results have been obtained for loss functions with/without Lipschitz continuity and for learning tasks including KRR [25], [100] and SVM [7], [58], [74]. Apart from supervised learning with random features, results on randomized nonlinear component analysis refer to [6], random features with matrix sketching [104], doubly stochastic gradients scheme [78], statistical consistency [105], [106].

5.2.1 Assumptions

Before we detail these theoretical results, we summarize the standard assumptions imposed in existing work. Some assumptions are technical, and thus familiarity with statistical learning theory (see Section 2.1) would be helpful. We organize these assumptions in four categories as shown in Figure 3, including i) the existence of f_ρ (Assumption 1) and its stronger version (Assumption 8); ii) quality of random features (Assumptions 2, 6, 7); iii) noise conditions (Assumptions 3, 9, 10); iv) eigenvalue decay (Assumptions 4, 5).

We first state three basic assumptions, which are needed in all of the (regression) results to be presented.

Assumption 1 (Existence [100], [107]). *$f_\rho \in \mathcal{H}$.*

Assumption 2 (Random features are bounded and continuous [100]). *For the shift-invariant kernel k, we assume that $\varphi(\boldsymbol{\omega}^\top\boldsymbol{x})$ in Eq. (6) is continuous in both variables and bounded, i.e., there exists $\kappa \geq 1$ such that $|\varphi(\boldsymbol{\omega}^\top\boldsymbol{x})| < \kappa$ for all $\boldsymbol{x} \in \mathcal{X}$ and $\boldsymbol{\omega} \in \mathbb{R}^d$.*

Assumption 3 (Bernstein's condition [108], [109]). *For any $\boldsymbol{x} \in \mathcal{X}$, we assume $\mathbb{E}\left[|y|^b \mid \boldsymbol{x}\right] \leq \frac{1}{2} b! \varsigma^2 B^{b-2}$ when $b \geq 2$.*

Assumption 3 is satisfied when y is bounded or sub-Gaussian.

The above three assumptions are needed in all theoretical results for regression presented in this section, so we omit them when stating these results. We next introduce several additional assumptions, which are needed in some of the theoretical results.

Eigenvalue Decay Assumptions: The following assumption, which characterizes the "size" of the RKHS $\mathcal{H}$ of interest, is often discussed in learning theory.

Assumption 4 (Eigenvalue decays [94]). *A kernel matrix $\boldsymbol{K}$ admit the following three types of eigenvalue decays: 1)* Geometric/exponential decay*: $\lambda_i(\boldsymbol{K}) \propto n\exp(-i^{1/c})$, which leads to $d^\lambda_{\boldsymbol{K}} \lesssim \log(R_0/\lambda)$; 2)* Polynomial decay*: $\lambda_i(\boldsymbol{K}) \propto n i^{-2a}$, which implies $d^\lambda_{\boldsymbol{K}} \lesssim (1/\lambda)^{1/2a}$; 3)* Harmonic decay*: $\lambda_i(\boldsymbol{K}) \propto n/i$, which results in $d^\lambda_{\boldsymbol{K}} \lesssim (1/\lambda)$.*

Generally, for stationary kernels, a smaller RKHS indicates a faster eigenvalue decay, of which functions are smooth enough to achieve a good prediction performance. It can be linked to the integral operator [107], [108] characterizing the hypothesis space, defined as $\Sigma : L^2_{\rho_\mathcal{X}} \to L^2_{\rho_\mathcal{X}}$ such that $(\Sigma g)(\boldsymbol{x}) = \int_\mathcal{X} k(\boldsymbol{x},\boldsymbol{x}')g(\boldsymbol{x}')\mathrm{d}\rho_\mathcal{X}(\boldsymbol{x}'), \forall g \in L^2_{\rho_\mathcal{X}}$. It is clear that the operator Σ is self-adjoint, positive definite, and trace-class when $k(\cdot,\cdot)$ is continuous. In particular, the decay rate of the spectrum of Σ quantifies the capacity of the hypothesis space in which we search for the solution by the following assumption.

Table 3
Comparison of convergence rates and required random features for kernel approximation error.

Metric	Results	Convergence rate	Upper bound of $\lvert\mathcal{S}\rvert$	Required random features s
$\lVert k-\tilde{k}\rVert_\infty$	Theorem 1 ([4], [24])	$\mathcal{O}_p\left(\lvert\mathcal{S}\rvert\sqrt{\frac{\log s}{s}}\right)$	$\lvert\mathcal{S}\rvert\leq\Omega\left(\sqrt{\frac{s}{\log s}}\right)$	$s\geq\Omega\left(d\epsilon^{-2}\log\frac{\lvert\mathcal{S}\rvert}{\epsilon}\right)$
	Theorem 1 in [98]	$\mathcal{O}_p\left(\sqrt{\frac{\log\lvert\mathcal{S}\rvert}{s}}\right)$	$\lvert\mathcal{S}\rvert\leq\Omega(s^c)$[1]	$s\geq\Omega\left(d\epsilon^{-2}\log\lvert\mathcal{S}\rvert\right)$
	Theorem 1 in [99] (Gaussian kernels)	$\mathcal{O}_p\left(\sqrt{\frac{\log\lvert\mathcal{S}\rvert}{s}}\right)$	$\lvert\mathcal{S}\rvert\leq\Omega(s^c)$	$s\geq\Omega\left(\epsilon^{-2}\log\lvert\mathcal{S}\rvert\right)$
$\lVert k-\tilde{k}\rVert_{L^r}\ (1\leq r<\infty)$	Corollary 2 in [98]	$\mathcal{O}_p\left(\lvert\mathcal{S}\rvert^{\frac{2d}{r}}\sqrt{\frac{\log\lvert\mathcal{S}\rvert}{s}}\right)$	$\lvert\mathcal{S}\rvert\leq\Omega\left((\frac{s}{\log s})^{\frac{r}{4d}}\right)$	$s\geq\Omega\left(d\epsilon^{-2}\log\lvert\mathcal{S}\rvert\right)$
$\lVert k-\tilde{k}\rVert_{L^r}\ (2\leq r<\infty)$	Theorem 3 in [98]	$\mathcal{O}_p\left(\lvert\mathcal{S}\rvert^{\frac{2d}{r}}\sqrt{\frac{1}{s}}\right)$	$\lvert\mathcal{S}\rvert\leq\Omega\left(s^{\frac{r}{4d}}\right)$	$s\geq\Omega\left(d\epsilon^{-2}\log\lvert\mathcal{S}\rvert\right)$
Δ-spectral approximation	Theorem 7 in [75]	$\mathcal{O}_p\left(\sqrt{\frac{n_\lambda}{s}}\right)$	-	$s\geq\Omega(n_\lambda\log d_{\boldsymbol{K}}^\lambda)$
	Theorem 5.4 in [81] (Gaussian kernels)	$\mathcal{O}_{\text{RFF/ORF}}\left(\frac{1}{s\lambda^2}\right)$	-	$s\geq\Omega(n^{2\alpha})$
	Lemma 6 in [75]	$\mathcal{O}_q\left(\sqrt{\frac{d_{\boldsymbol{K}}^\lambda}{s}}\right)$	-	$s\geq\Omega(d_{\boldsymbol{K}}^\lambda\log d_{\boldsymbol{K}}^\lambda)$
(Δ_1,Δ_2)-spectral approximation	Theorem 2 in [70]	$\mathcal{O}_{\text{LP}}\left(\sqrt{\frac{n_\lambda}{s}}\right)$[2]	-	$s\geq\Omega(n_\lambda\log d_{\boldsymbol{K}}^\lambda)$

[1] c is some constant satisfying $0<c<1$.

[2] LP denotes that $\{\boldsymbol{\omega}_i\}_{i=1}^s$ are obtained by RFF and then are quantized to a Low-Precision b-bit representation; see [70].

- i) regression: $f_\rho\in\mathcal{H}$ (**Ass. 1**) $\Leftarrow$ source condition (**Ass. 8**)
- ii) quality of random features
  - bounded and continuous (**Ass. 2**)
  - $q(\boldsymbol{\omega})$-sampling: compatibility condition (**Ass. 6**) $\Leftarrow$ optimized distribution (**Ass. 7**)
- iii) noise condition
  - regression: boundedness on y (**Ass. 3**)
  - classification: Massart's low noise condition (**Ass. 9**) $\Leftarrow$ separation condition (**Ass. 10**)
- iv) eigenvalue decays assumption (**Ass. 4**)
  - exponential decay
  - polynomial decay and harmonic decay $\Leftrightarrow$ capacity condition (**Ass. 5**)

Figure 3. Relationship between the needed assumptions. The notation $A\Leftarrow B$ means that B is a stronger assumption than A.

Assumption 5 (Capacity condition [107], [110]). *There exist $Q>0$ and $\gamma\in[0,1]$ such that for any $\lambda>0$, we have*

$$\mathcal{N}(\lambda):=\operatorname{tr}\left((\Sigma+\lambda I)^{-1}\Sigma\right)\leq Q^2\lambda^{-\gamma}\,. \tag{19}$$

The effective dimension $\mathcal{N}(\lambda)$ [93] measures the "size" of the RKHS, and is in fact the operator form of $d_{\boldsymbol{K}}^\lambda$ in Eq. (17). Assumption 5 holds if the eigenvalues λ_i of Σ decay as $i^{-1/\gamma}$, which corresponds to the eigenvalue decay of $\boldsymbol{K}$ in Assumption 4 with $\gamma:=1/(2a)$ [111]. The case $\gamma=0$ is the more benign situation, whereas $\gamma=1$ is the worst case.

Quality of Random Features: Here we introduce several technical assumptions on the quality of random features. The leverage score in Eq. (16) admits the operator form

$$\mathcal{F}_\infty(\lambda):=\sup_{\boldsymbol{\omega}}\left\|(\Sigma+\lambda I)^{-1/2}\varphi(\boldsymbol{x})\right\|_{L^2_{\rho_\mathcal{X}}}^2,\ \forall\lambda>0\,,$$

which is also called as the *maximum random features dimension* [100]. By defintion we always have $\mathcal{N}(\lambda)\leq\mathcal{F}_\infty(\lambda)$. Roughly speaking, when the random features are "good", it is easy to control their leverage scores in terms of the decay of the spectrum of Σ. Further, fast learning rates using fewer random features can be achieved if the features are *compatible* with the data distribution in the following sense.

Assumption 6 (Compatibility condition [100]). *With the above definition of $\mathcal{F}_\infty(\lambda)$, assume that there exist $\varrho\in[0,1]$, and $F>0$ such that $\mathcal{F}_\infty(\lambda)\leq F\lambda^{-\varrho},\forall\lambda>0$.*

It always holds that $\mathcal{F}_\infty(\lambda)\leq\kappa^2\lambda^{-1}$ when z is uniformly bounded by κ. So the worst case is $\varrho=1$, which means that the random features are sampled in a problem independent way. The favorable case is $\varrho=\gamma$, which means that $\mathcal{N}(\lambda)\leq\mathcal{F}_\infty(\lambda)\leq\mathcal{O}(n^{-\alpha\gamma})$. In [7], the authors consider the following assumption.

Assumption 7 (Optimized distribution [7]). *The feature mapping $z(\boldsymbol{\omega},\boldsymbol{x})$ is called* optimized *if there is a small constant λ_0 such that for any $\lambda\leq\lambda_0$, $\mathcal{F}_\infty(\lambda)\leq\mathcal{N}(\lambda)=\sum_{i=1}^\infty\frac{\lambda_i(\Sigma)}{\lambda_i(\Sigma)+\lambda}$.*

Under the previous definitions, Assumption 7 holds only when $\mathcal{F}_\infty(\lambda)=\mathcal{N}(\lambda)$. This assumption is stronger than the compatibility condition in Assumption 6. Note that Assumption 7 is satisfied when sampling from $q(\boldsymbol{\omega})$.

Source condition on f_ρ: The following assumption states that f_ρ has some desirable regularity properties.

Assumption 8 (Source condition [100], [112]). *There exist $1/2\leq r\leq 1$ and $g\in L^2_{\rho_\mathcal{X}}$ such that $f_\rho(\boldsymbol{x})=(\Sigma^r g)(\boldsymbol{x})$ almost surely.*

Since Σ is a compact positive operator on $L^2_{\rho_\mathcal{X}}$, its r-th power

Table 4
Comparison of learning rates and required random features for expected risk with the squared loss function.

sampling scheme	Results	key assumptions	eigenvalue decays	λ	learning rates	required s
$\{\boldsymbol{\omega}_i\}_{i=1}^s \sim p(\boldsymbol{\omega})$	[100, Theorem 1]	-	-	$n^{-\frac{1}{2}}$	$\mathcal{O}_p\left(n^{-\frac{1}{2}}\right)$	$s \geq \Omega(\sqrt{n}\log n)$
	[100, Theorem 2]	source condition	i^{-2t}	$n^{-\frac{2t}{1+4rt}}$	$\mathcal{O}_p\left(n^{-\frac{4rt}{1+4rt}}\right)$	$s \geq \Omega(\frac{2t+2r-1}{1+4rt}\log n)$
			$1/i$	$n^{-\frac{1}{2r+1}}$	$\mathcal{O}_p\left(n^{-\frac{2r}{2r+1}}\right)$	$s \geq \Omega(n^{\frac{2r}{2r+1}}\log n)$
	[25, Corollary 2]	-	$e^{-\frac{1}{c}i}$	$n^{-\frac{1}{2}}$	$\mathcal{O}_p\left(n^{-\frac{1}{2}}\right)$	$s \geq \Omega(\sqrt{n}\log n)$
			i^{-2t}	$n^{-\frac{1}{2}}$	$\mathcal{O}_p\left(n^{-\frac{1}{2}}\right)$	$s \geq \Omega(\sqrt{n}\log n)$
			$1/i$	$n^{-\frac{1}{2}}$	$\mathcal{O}_p\left(n^{-\frac{1}{2}}\right)$	$s \geq \Omega(\sqrt{n}\log n)$
$\{\boldsymbol{\omega}_i\}_{i=1}^s \sim q(\boldsymbol{\omega})$	[100, Theorem 3]	source condition; compatibility condition	i^{-2t}	$n^{-\frac{2t}{1+4rt}}$	$\mathcal{O}_q\left(n^{-\frac{4rt}{1+4rt}}\right)$	$s \geq \Omega(\frac{\varrho+(2r-1)(2t+1-2t\varrho)}{1+4rt}\log n)$
			$1/i$	$n^{-\frac{1}{2r+1}}$	$\mathcal{O}_q\left(n^{-\frac{2r}{2r+1}}\right)$	$s \geq \Omega(n^{\frac{2r}{2r+1}}\log n)$
	[25, Corollary 1]	optimized distribution	$e^{-\frac{1}{c}i}$	$n^{-\frac{1}{2}}$	$\mathcal{O}_q\left(n^{-\frac{1}{2}}\right)$	$s \geq \Omega(\log^2 n)$
			i^{-2t}	$n^{-\frac{1}{2}}$	$\mathcal{O}_q\left(n^{-\frac{1}{2}}\right)$	$s \geq \Omega(n^{1/(2t)}\log n)$
			$1/i$	$n^{-\frac{1}{2}}$	$\mathcal{O}_q\left(n^{-\frac{1}{2}}\right)$	$s \geq \Omega(\sqrt{n}\log n)$

Σ^r is well defined for any $r > 0$.[6] Assumption 8 imposes a form of regularity/sparsity of f_ρ, which requires the expansion of f_ρ on the basis given by the integral operator Σ. Note that this assumption is more stringent than the existence of f_ρ in $\mathcal{H}$. The latter is equivalent to Assumption 8 with $r = \frac{1}{2}$ (the worst case), in which case $f_\rho \in \mathcal{H}$ need not have much regularity/sparsity.

Noise Condition: The following two assumptions on noise are considered in random features for classification.

Assumption 9 (Massart's low noise condition [7], [115]). *There exists $V \geq 2$ such that $\left|\mathbb{E}_{(\boldsymbol{x},y)\sim\rho}[y|\boldsymbol{x}]\right| \geq 2/V$.*

Assumption 10 (Separation condition [7]). *The points in $\mathcal{X}$ can be collected into two sets according to their labels as follows*

$$X_1 := \{\boldsymbol{x} \in \mathcal{X} : \mathbb{E}[y|\boldsymbol{x}] > 0\}, \quad X_{-1} := \{\boldsymbol{x} \in \mathcal{X} : \mathbb{E}[y|\boldsymbol{x}] < 0\}.$$

The distance of a point $\boldsymbol{x} \in X_{\pm 1}$ to the set $X_{\mp 1}$ is denoted by $\Delta(\boldsymbol{x})$. We say that the data distribution satisfies a separation condition if there exists $\Delta > 0$ such that $\rho_X(\Delta(\boldsymbol{x}) < c) = 0$.

The above two assumptions, both controlling the noise level in the labels, can be cast as special cases of Tsybakov's low noise assumption [115].

5.2.2 Squared loss in KRR

In this section, we review theoretical results on the generalization properties of KRR with squared loss and random features, for both the $p(\boldsymbol{\omega})$-sampling (data-independent) and $q(\boldsymbol{\omega})$-sampling (data-dependent) settings. Table 4 summarizes these results for the excess risk in terms of the key assumptions imposed, the learning rates, and the required number of random features.

We begin with the remarkable result by Rudi and Rosasco [100]. They are among the first to show that under some mild assumptions and appropriately chosen parameters, $\Omega(\sqrt{n}\log n)$ random features suffice for KRR to achieve minimax optimal rates.

6. A more general condition ($r > 0$) is often considered in approximation theory; see [113], [114].

Theorem 2 (Generalization bound; Theorem 3 in [100]). *Suppose that Assumption 8 (source condition) holds with $r \in [\frac{1}{2}, 1]$, Assumption 6 (compatibility) holds with $\varrho \in [0, 1]$, and Assumption 5 (capacity) holds with $\gamma \in [0, 1]$. Assume that $n \geq n_0$ and choose $\lambda := n^{\frac{1}{2r+\gamma}}$. If the number of random features satisfies*

$$s \geq c_0 n^{\frac{\alpha+(2r-1)(1+\gamma-\alpha)}{2r+\gamma}} \log\frac{108\kappa^2}{\lambda\delta},$$

then the excess risk of $\tilde{f}_{\boldsymbol{z},\lambda}$ can be upper bounded as

$$\mathcal{E}\left(\tilde{f}_{\boldsymbol{z},\lambda}\right) - \mathcal{E}\left(f_\rho\right) = \left\|\tilde{f}_{\boldsymbol{z},\lambda} - f_\rho\right\|_{L^2_{\rho_\mathcal{X}}}^2 \leq c_1 \log^2\frac{18}{\delta} n^{-\frac{2r}{2r+\gamma}},$$

where c_0, c_1 are constants independent of (n, λ, δ), and n_0 does not depends on n, λ, f_ρ, or ρ.

Theorem 2 unifies several results in [100] that impose different assumptions. If the compatibility condition is replaced by the stronger Assumption 7 (optimized distribution), satisfied by $q(\boldsymbol{\omega})$-sampling, the work [25] derives an improved bound that is the sharpest to date by spectral approximation. Below we state a general result from [25] that covers both $p(\boldsymbol{\omega})$- and $q(\boldsymbol{\omega})$-sampling.

Theorem 3 (Theorem 1 in [25]). *Suppose that the regularization parameter λ satisfies $0 \leq n\lambda \leq \lambda_1$. We consider two sampling schemes.*

- $\{\boldsymbol{\omega}_i\}_{i=1}^s \sim p(\boldsymbol{\omega})$*: if* $s \geq (5z_0^2/\lambda)\log(16d_{\boldsymbol{K}}^\lambda/\delta)$ *and* $|z(\boldsymbol{\omega}, \boldsymbol{x})| \leq z_0$,
- $\{\boldsymbol{\omega}_i\}_{i=1}^s \sim q(\boldsymbol{\omega})$*: if* $s \geq 5d_{\boldsymbol{K}}^\lambda \log\left(16d_{\boldsymbol{K}}^\lambda/\delta\right)$,

then for $0 < \delta < 1$, with probability $1 - \delta$, the excess risk of $\tilde{f}_{\boldsymbol{z},\lambda}$ can be upper bounded as

$$\left\|\tilde{f}_{\boldsymbol{z},\lambda} - f_\rho\right\|_{L^2_{\rho_\mathcal{X}}}^2 \leq 2\lambda + \mathcal{O}(1/\sqrt{n}) + \mathcal{E}\left(f_{\boldsymbol{z},\lambda}\right) - \mathcal{E}\left(f_\rho\right), \quad (20)$$

where we recall that $\mathcal{E}\left(f_{\boldsymbol{z},\lambda}\right) - \mathcal{E}\left(f_\rho\right)$ is the excess risk of standard KRR with an exact kernel (see Section 2).

Further, a sharper convergence rate can be achieved if the local Rademacher complexity technique [116] is used, see [117] for

details. Besides, Carratino et al. [118] extend the result of [100] to the setting where KRR is solved by stochastic gradient descent (SGD). They show that under the basic Assumptions 1-3 and some mild conditions for SGD, $\Omega(\sqrt{n})$ random features suffice to achieve the minimax learning rate $\mathcal{O}(n^{-1/2})$. Wang [119] gives the out-of-sample bound $\mathcal{E}(\tilde{f}_{\boldsymbol{z},\lambda}) - \mathcal{E}(f_{\boldsymbol{z},\lambda}) \leq \mathcal{O}\left(1/(s\lambda)\right)$ under the boundedness of the randomized feature map (which is weaker than Assumption 2). If we choose $\lambda := n^{-1/2}$, then $\Omega(n)$ random features are sufficient to ensure an $\mathcal{O}(n^{-1/2})$ rate in the out-of-sample bound.

5.2.3 Lipschitz continuous loss function

In this section, we consider loss functions ℓ that are Lipschitz continuous. Examples include the hinge loss in SVM and the cross-entropy loss in kernel logistic regression. Table 5 summarizes several existing results for such loss functions in terms of the learning rate and the required number of random features. We briefly discuss these results below and refer the readers to the cited work for the precise theorem statements.

If $\{\boldsymbol{\omega}_i\}_{i=1}^s \sim p(\boldsymbol{\omega})$, i.e., under the standard RFF setting with data-independent sampling, we have the following results.

- Theorem 1 in [58] shows that the excess risk converges at a certain $\mathcal{O}(n^{-1/2})$ rate with $\Omega(n \log n)$ random features.
- Corollary 4 in [25] shows that with $\lambda \in \mathcal{O}(1/n)$ and $\Omega\big((1/\lambda)\log d_{\boldsymbol{K}}^{\lambda}\big)$ random features, the excess risk of $\tilde{f}_{\boldsymbol{z},\lambda}$ can be upper bounded by

$$\mathcal{E}(\tilde{f}_{\boldsymbol{z},\lambda}) - \mathcal{E}\left(f_\rho\right) \leq \mathcal{O}\left(1/\sqrt{n}\right) + \mathcal{O}(\sqrt{\lambda})\,.$$

  The above bound scales with $\sqrt{\lambda}$, which is different from the bound in Eq. (20) for the squared loss. Therefore, for Lipschitiz continuous loss functions, we need to choose a smaller regularization parameter $\lambda \in \mathcal{O}(1/n)$ to achieve the same $\mathcal{O}(n^{-1/2})$ convergence rate. Also note that as before we can bound $d_{\boldsymbol{K}}^{\lambda}$ under the three types of eigenvalue decay.

If $\{\boldsymbol{\omega}_i\}_{i=1}^s \sim q(\boldsymbol{\omega})$, i.e., under the data-dependent sampling setting, we have the following results.

- For SVM with random features, under the optimized distribution in Assumption 7 and the low noise condition in Assumption 9, Theorem 1 in [7] provides bounds on the learning rates and the required number of random features. This result is improved in [7, Theorem 2] if we consider the stronger separation condition in Assumption 10. Details can be found in Table 5.
- In Section 4.5 in [74] and Corollary 3 in [25], it is shown that if Assumption 7 holds, then the excess risk of $\tilde{f}_{\boldsymbol{z},\lambda}$ converges at an $\mathcal{O}(n^{-1/2})$ rate with $\Omega(d_{\boldsymbol{K}}^{\lambda}\log d_{\boldsymbol{K}}^{\lambda})$ random features, if we choose $\lambda \in \mathcal{O}(1/n)$.

There is an abnormal but common experiment phenomenon on kernel approximation and risk generalization, that is, a higher kernel approximation quality does not always translate to better generalization performance, see the discussion in [21], [70], [75]. Understanding this inconsistency between approximation quality and generalization performance is an important open problem in this topic. Here we present a preliminary result for KRR: a better approximation quality cannot guarantee a lower generalization risk, see Proposition 1 as below, with proof deferred to Appendix.

Proposition 1. *Given the target function f_ρ and the original kernel matrix $\boldsymbol{K}$, consider two random features based algorithms* `A1` *and* `A2` *yielding two approximated kernel matrices $\widetilde{\boldsymbol{K}}_1$ and $\widetilde{\boldsymbol{K}}_2$, and their respective KRR estimators $\tilde{f}_{\boldsymbol{z},\lambda}^{(\mathtt{A1})}$ and $\tilde{f}_{\boldsymbol{z},\lambda}^{(\mathtt{A2})}$. Then for a new sample $\boldsymbol{x}$, even if $\|\boldsymbol{K} - \widetilde{\boldsymbol{K}}_1\| \leq \|\boldsymbol{K} - \widetilde{\boldsymbol{K}}_2\|$ holds in some norm metric, there exists one case for the excess risk such that*

$$\mathcal{E}[\tilde{f}_{\boldsymbol{z},\lambda}^{(\mathtt{A1})}(\boldsymbol{x})] - \mathcal{E}[f_\rho(\boldsymbol{x})] \geq \mathcal{E}[\tilde{f}_{\boldsymbol{z},\lambda}^{(\mathtt{A2})}(\boldsymbol{x})] - \mathcal{E}[f_\rho(\boldsymbol{x})]\,.$$

Remark: Our proof is geometric by constructing a counter-example. It requires that the kernel admits (at least) polynomial decay, which holds for the common-used Gaussian kernel and could be further relaxed for the existence of the proof.

6 Experiments

In this section, we empirically evaluate the kernel approximation and classification performance of representative random features algorithms on several benchmark datasets. All experiments are implemented in MATLAB and carried out on a PC with Intel® i7-8700K CPU (3.70 GHz) and 64 GB RAM. The source code of our implementation can be found in http://www.lfhsgre.org.

6.1 Experimental settings

We choose the popular Gaussian kernel, zero/first-order arc-cosine kernels, and polynomial kernel evaluated on several medium/large scale benchmark datasets. Table 6 gives an overview of these datasets including the number of feature dimension, training samples, test data, training/test split, and the normalization scheme. We use $\|\boldsymbol{K} - \widetilde{\boldsymbol{K}}\|_{\mathrm{F}}/\|\boldsymbol{K}\|_{\mathrm{F}}$ as the error metric for kernel approximation. To compute the approximation error, we randomly sample 1,000 data points to construct the sub-feature matrix and the sub-kernel matrix. For the subsequent classification task, the random feature mappings are used with two classifiers: the ridge linear regression (abbreviated as lr) with the squared loss, and the liblinear algorithm [120] (a linear classifier with the hinge loss). All experiments are repeated 10 times and we report the average approximation error, average classification accuracy with their respective standard deviations as well as the time cost for generating random features. More detailed description of these datasets and experimental settings can be found in Appendix.

6.2 Results for the Gaussian Kernel

6.2.1 Results on non-image benchmark datasets

Here we test various random features based algorithms, including RFF [4], ORF [18], SORF [18], ROM [51], Fastfood [47], QMC [52], SSF [53], GQ [20], LS-RFF [25] for kernel approximation and then combine these algorithms with lr/liblinear for classification on eight non-image benchmark datasets. Here we summarize the best performing algorithm on each dataset in terms of the approximation quality in Table 7, where we distinguish the small s case (i.e., $s = 2d$ or $s = 4d$) and the large s case (i.e., $s = 16d$ or $s = 32d$). The notation "-" therein means that there is no *statistically significant* difference in the performance of most algorithms. Nevertheless, most of algorithms obtain the similar performance on the test accuracy.

In terms of approximation error, we find that SSF, ORF, and QMC achieve promising approximation performance in most cases. Recall that the goal of using random features is to find a finite-dimensional (embedding) Hilbert space to approximate the original infinite-dimensional RKHS so as to preserve the inner product. To achieve this goal, SSF, QMC, and ORF are based on a similar principle, namely, generating random features that are as independent/complete as possible to reduce the randomness in sampling. Regarding to SSF, we find that SSF performs well under the small s case, but the significant improvement does not hold

Table 5
Comparison of learning rates and required random features for expected risk with a Lipschitz continuous loss function.

sampling scheme	Results	key assumptions	eigenvalue decay	λ	learning rates	required s
$\{\boldsymbol{\omega}_i\}_{i=1}^s \sim p(\boldsymbol{\omega})$	[58, Theorem 1]	-	-	-	$\mathcal{O}_p\left(n^{-\frac{1}{2}}\right)$	$s \geq \Omega(n \log n)$
	[25, Corollary 4]	-	$e^{-\frac{1}{c}i}$	$\frac{1}{n}$	$\mathcal{O}_p\left(n^{-\frac{1}{2}}\right)$	$s \geq \Omega(n \log \log n)$
			i^{-2t}	$\frac{1}{n}$	$\mathcal{O}_p\left(n^{-\frac{1}{2}}\right)$	$s \geq \Omega(n \log n)$
			$1/i$	$\frac{1}{n}$	$\mathcal{O}_p\left(n^{-\frac{1}{2}}\right)$	$s \geq \Omega(n \log n)$
$\{\boldsymbol{\omega}_i\}_{i=1}^s \sim q(\boldsymbol{\omega})$	[7, Theorem 1]	low noise condition	$e^{-\frac{1}{c}i}$	$\frac{1}{n}$	$\mathcal{O}_q(\frac{1}{n}\log^{c+2} n)$	$s \geq \Omega(\log^c n \log\log^c n)$
		optimized distribution	i^{-2t}	$n^{-\frac{t}{1+t}}$	$\mathcal{O}_q\left(n^{-\frac{t}{1+t}}\log n\right)$	$s \geq \Omega(n^{\frac{1}{1+t}}\log n)$
			$1/i$	$\frac{1}{n}$	$\mathcal{O}_q\left(n^{-\frac{1}{2}}\right)$	$s \geq \Omega(n \log n)$
	[7, Theorem 2]	separation condition optimized distribution	$e^{-\frac{1}{c}i}$	n^{-2c^2}	$\mathcal{O}_q(\frac{1}{n}\log^{2c+1} n \log\log n)$	$s \geq \Omega(\log^{2c} n \log\log n)$
	[74, Section 4.5] [25, Corollary 3]	optimized distribution	$e^{-\frac{1}{c}i}$	$\frac{1}{n}$	$\mathcal{O}_q\left(n^{-\frac{1}{2}}\right)$	$s \geq \Omega(\log^2 n)$
			i^{-2t}	$\frac{1}{n}$	$\mathcal{O}_q\left(n^{-\frac{1}{2}}\right)$	$s \geq \Omega(n^{1/(2t)}\log n)$
			$1/i$	$\frac{1}{n}$	$\mathcal{O}_q\left(n^{-\frac{1}{2}}\right)$	$s \geq \Omega(n \log n)$

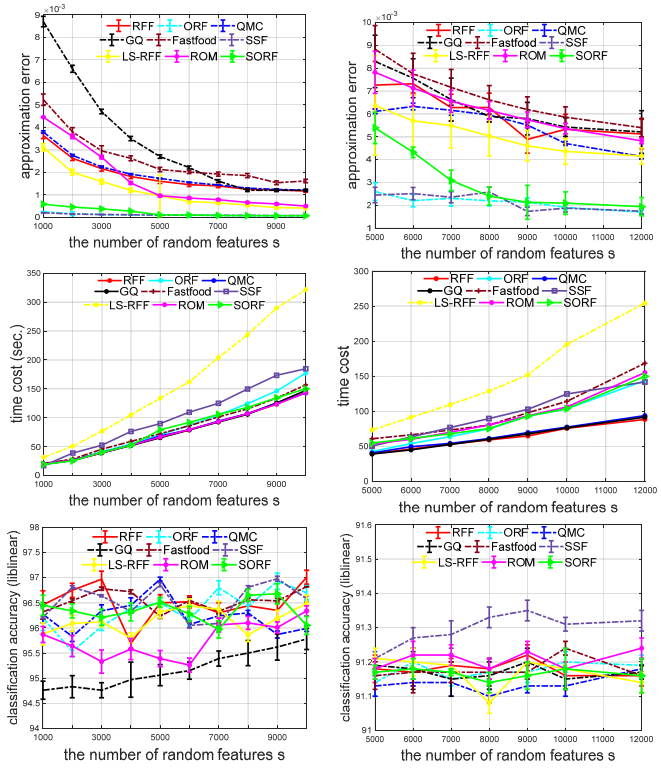
Table 6
Dataset statistics.

datasets	d	#traing	#test	random split	scaling
ijcnn1	22	49,990	91,701	no	-
EEG	14	7,490	7,490	yes	mapstd
cod-RNA	8	59,535	157,413	no	mapstd
covtype	54	290,506	290,506	yes	minmax
magic04	10	9,510	9,510	yes	minmax
letter	16	12,000	6,000	no	minmax
skin	3	122,529	122,529	yes	minmax
a8a	123	22,696	9,865	no	-
MNIST	784	60,000	10,000	no	minmax
CIFAR-10	3072	50,000	10,000	no	-
MNIST-8M	784	8,100,000	10,000	no	-

Table 7
Results statistics on several datasets. The best algorithm on each dataset is given in two cases: low dimensional (i.e., $s = 2d, 4d$) and high dimensional (i.e., $s = 16d, 32d$) according to approximation quality. The notation "-" means that there is no *statistically significant* difference in the performance of most algorithms.

datasets	approximation	
	small s	large s
ijcnn1	SSF	SORF, QMC, ORF
EEG	SSF	ORF
cod-RNA	SSF	-
covtype	ORF	-
magic04	SSF	SSF, ORF, QMC, ROM
letter	SSF	SSF, ORF
skin	SSF, ROM	QMC
a8a	-	-

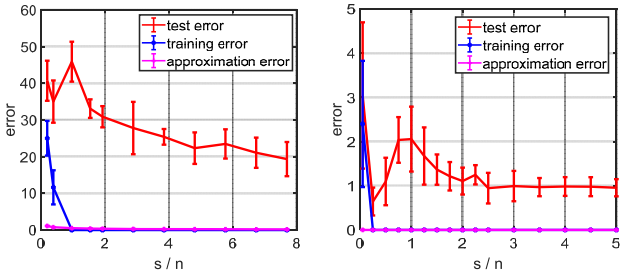
for the large s case. This might be because, a few points can be adequate in SSF, additional points (i.e., a larger s) may have a small marginal benefit in variance reduction under the large s setting. Consequently, the approximation error of SSF sometimes stays almost the same with a larger number of random features. QMC and



(a) MNIST (b) CIFAR10

Figure 4. Approximation error, time cost, and test accuracy of various algorithms with liblinear on two image classification datasets.

ORF seek for variance reduction on random features. Nevertheless, they often work well in the large s case. As demonstrated by the expression for variance of ORF [18] and convergence rate in QMC [52], this theoretical result is consistent with the numerical performance of ORF and QMC, which may explain the reason why they work better in a large s setting than a small s case. Besides, results on arc-cosine kernels and polynomial kernels, and evaluation on *MNIST 8M* [121] can be found in Appendix.



(a) sonar (low-dimensioanl) (b) MNIST (high-dimensional)

Figure 5. Training error, test error, and approximation error of random features regression with $\lambda = 10^{-8}$ on the *sonar* dataset with $n = 208, d = 60$ and the sub-set of MNIST (class 1 versus class 2) with $n = 200, d = 784$.

6.2.2 *Classification results on* MNIST *and* CIFAR10

Here we consider the MNIST and CIFAR10 datasets, on which we test these random features based algorithms for kernel approximation and then combine these algorithms with liblinear for image classification. In our experiment, we use the Gaussian kernel[7], whose kernel width ς is tuned by 5-fold cross validation over the grid $\varsigma = [0.01, 0.1, 1, 10, 100]$. For the MNIST database, we directly use the original 784-dimensional feature as the data. For better performance on the CIFAR10 dataset, we use VGG16 with batch normalization [123] pre-trained on ImageNet [124] as a feature extractor. We fine-tune this model on the CIFAR10 dataset with 240 epochs and a mini-batch size 64. The learning rate is initialized at 0.1 and then divided by 10 at the 120-th, 160-th, and 200-th epochs. For each color image, a 4096 dimensional feature vector is obtained from the output of the first fully-connected layer in this fine-tuned neural network.

Figure 4(a) shows the approximation error, the time cost (sec.), and the classification accuracy by liblinear across a range of $s = 1000$ to $s = 10,000$ random features on the MNIST database. We find that ORF and SSF yield the best approximation quality. Despite that most algorithms achieve different approximation errors, there is no significant difference in the test accuracy, which corresponds to the results on non-image datasets. Similar results are observed on the CIFAR10 dataset with $s = 5000$ to $s = 12,000$ random features; see Figure 4(b). Note that most algorithms take the similar time cost on generating random features except for the data-dependent algorithm LS-RFF. Several structured based approaches (e.g., Fastfood, SORF, ROM) do not achieve significant reduction on time cost due to the relatively inefficient Matlab built-in function to implement the Walsh-Hadamard transform.

7 Trends: High-dimensional Random Features in Over-parameterized Settings

In the previous sections, we review random features based algorithms and their theoretical results, that works under a fixed d setting with $s \ll n$. Random features based approaches are simple in formulation but enjoy nice empirical validations and theoretical guarantees in kernel approximation and generalization properties. Recently, analysis of over-parameterized models [16], [125], [126], [127], [128] has attracted a lot of attention in learning theory, partly due to the observation of several intriguing phenomena, including capability of fitting random labels, strong generalization performance of overfitted classifiers [129] and double descent in the test error curve [130], [131]. Moreover, Belkin et al. [130], [132] point out that the above phenomena are not unique to deep networks but also exist in random features and random forests. In Figure 5, we report the empirical training error, the test error, and the kernel approximation error of random features regression as a function of s/n on the *sonar* dataset and the *MNIST* dataset [133]. Even with n, d, s only in the hundreds, we can still observe that as s increases, the training error reduces to zero and the approximation error monotonously decreases. However, the test error exhibits double descent, i.e., a phase transition at the *interpolation threshold*: moving away from this threshold on both sides trends to reduce the generalization error. This is somewhat striking as it goes against the conventional wisdom on *bias-variance trade-off* [134]: predictors that generalize well should trade off the model complexity against training data fitting.

The above observations have motivated researchers to build on the elegant theory of random features to provide an analysis of neural networks in the over-parameterized regime. To be specific, RFF can be regarded as a two-layer (large-width) neural network, where the weights in the first layer are chosen randomly/fixed and only the output layer is optimized. This is a typical over-parameterized model if we take $s \gg n$. As such, two-layer neural networks in this regime are more amenable to theoretical analysis as compared to general arbitrary deep networks. This is a potentially fruitful research direction, and one hand, the optimization and generalization of such model have been studied in [16], [135] in deep learning theory. On the other hand, in order to explain the double descent curve of random features in over-parameterized regimes, we often work in a high dimensional setting, which is more subtle than classical results in standard settings, as indicated by recent random matrix theory (RMT) [136], [137], [138]. An intuitive example [139] is, $\|\boldsymbol{K} - \boldsymbol{Z}\boldsymbol{Z}^\top\|_{\mathrm{F}} \to 0$ always hold in low/high dimensions as $s \to \infty$ but $\|\boldsymbol{K} - \boldsymbol{Z}\boldsymbol{Z}^\top\|_2 \to 0$ does not hold for $n, d, s \to \infty$. Accordingly, in this section, we provide an overview on analysis of (high dimensional) random features in over-parameterized setting, especially on double descent. We remark upfront that the random features model on double descent is not the only way for analyzing DNNs. Many other approaches, with different points of views, have been proposed for deep learning theory, but they are out of scope of this survey.

7.1 Results on High Dimensional Random Features in Over-parameterized Setting

Here we briefly introduce the problem setting of high dimensional random features in over-parameterized regimes, and then discuss the techniques used in various studies.

In the basic setting, high dimensional random features often work with least squares regression setting in an asymptotic viewpoint, i.e., $n, d, s \to \infty$ with $d/n \to \psi_1 \in (0, \infty)$ and $s/n \to \psi_2 \in (0, \infty)$, in which overparameterization corresponds to $\psi_2 \geq 1$. The considered data generation model in the basic setting is quite simple. To be specific, the training data is collected in a matrix $\boldsymbol{X} \in \mathbb{R}^{n \times d}$, the rows of which are assumed to be drawn i.i.d from $\mathcal{N}(0, 1)$ or $\mathbb{S}^{d-1}(\sqrt{d})$. The labels are given by a linear ground truth corrupted by some independent additive Gaussian noise: $y_i = f_\rho(\boldsymbol{x}_i) + \varepsilon_i$, where $f_\rho(\boldsymbol{x}) = \langle \boldsymbol{x}, \boldsymbol{\zeta} \rangle$ for a fixed but unknown $\boldsymbol{\zeta}$ and $\varepsilon_i \sim \mathcal{N}(0, 1)$. The transformation matrix under this setting is often taken as the random Gaussian matrix with the

7. As indicated by [9], [122], (convolutional) NTK generally performs better than Gaussian kernel but it is still non-trivial to obtain a efficient random features mapping for (convolutional) NTK without much loss on prediction.

Table 8
Comparison of problem settings on analysis of high dimensional random features on double descent.

studies	metric	data generation				asymptotic?	result
		$\{\boldsymbol{x}_i\}_{i=1}^n$	f_ρ	activation function	$\boldsymbol{W}$		
[125, Theorem 7]	population risk	$\mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$	$\langle \boldsymbol{x}, \boldsymbol{\zeta} \rangle$	normalized	$\mathcal{N}(0, 1/d)$	✓	variance ↗ ↘
[140, Theorem 4]	population risk	$\mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$	$\langle \boldsymbol{x}, \boldsymbol{\zeta} \rangle$	bounded	$\mathcal{N}(0, 1/d)$	✓	variance ↗ ↘
[126, Theorem 2]	expected excess risk	$\mathbb{S}^{d-1}(\sqrt{d})$	$\langle \boldsymbol{x}, \boldsymbol{\zeta} \rangle$ + nonlinear [1]	bounded	Unif$(\mathbb{S}^{d-1}(\sqrt{d}))$	✓	variance, bias ↗ ↘
[141]	expected excess risk	$\mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$	$\langle \boldsymbol{x}, \boldsymbol{\zeta} \rangle$	ReLU	$\mathcal{N}(0, 1)$	✓	refined [2]
[142]	generalization error	$\mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$	$f(\langle \boldsymbol{x}, \boldsymbol{\zeta} \rangle)$	general	general	✓	↗ ↘
[143, Theorem 1]	generalization error	$\mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$	$\langle \boldsymbol{x}, \boldsymbol{\zeta} \rangle$	normalized	$\mathcal{N}(0, 1)$	✓	refined [2]
[144, Theorem 1]	generalization error	$\mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$	$\langle \boldsymbol{x}, \boldsymbol{\zeta} \rangle$	general	general	✓	↗ ↘
[145, Proposition 1]	generalization error	$\mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$	$\langle \boldsymbol{x}, \boldsymbol{\zeta} \rangle$	odd, bounded	sub-Gaussian	✓	↗ ↘
[146, Theorem 5.1]	expected excess risk	Gaussian	general	$[\cos(\cdot), \sin(\cdot)]$	$\mathcal{N}(0, 1)$	✗	↗ ↘
[139, Theorem 3]	generalization error	general	- [3]	$[\cos(\cdot), \sin(\cdot)]$	$\mathcal{N}(0, 1)$	✓	↗ ↘

[1] The nonlinear component is a centered isotropic Gaussian process indexed by $\boldsymbol{x} \in \mathbb{S}^{d-1}(\sqrt{d})$.
[2] A refined decomposition on variance is conducted by sources of randomness: "noise variance", "initialization variance", and "sampling variance" to possess each term [141] or their interpretations [143].
[3] It makes no assumptions on f_ρ but requires that test data "behave" statistically like the training data by concentrated random vectors.

ReLU activitation function (recall Eq. (6)). Current approaches employ various data generation schemes and assumptions to obtain a refined analysis beyond double descent under the basic setting. According to these criteria, we summarize the problem setting of various representative approaches in Table 8. In the next, we briefly review the conceptual and technical contributions of underlying approaches on high dimensional random features.

Belkin et al. [147] begin with an one-dimensional (noise-free) version of the random features model, and provide an asymptotic analysis to explain the double descent phenomenon. The subsequent work focuses on the standard random features model under different settings and assumptions. It is clear that, the presence of the nonlinear activation function $\sigma(\cdot)$ makes the random features model intractable to study the related (limiting) spectral distribution. Accordingly, the key issue in this topic mainly focuses on studying random matrices with nonlinear dependencies, e.g., how to disentangle the nonlinear function $\sigma(\cdot)$ by Gaussian equivalence conjecture. Hastie et al. [125] consider the basic setting endowed by a bounded activation function with a standardization condition, i.e., $\mathbb{E}[\sigma(t)] = 0$ and $\mathbb{E}[\sigma(t)^2] = 1$ for $t \sim N(0, 1)$. By establishing asymptotic results on resolvents of random block matrices from RMT, the limiting of the variance is theoretically demonstrated to be increasing for $\psi_2 \in (0, 1)$, decreasing for $\psi_2 \in (1, \infty)$, and diverging as $\psi_2 \to 1$.

In a similar spirit, Mei and Montanari [126] use RMT to study the spectral distribution of the Gram matrix $\boldsymbol{Z} = \sigma(\boldsymbol{X}\boldsymbol{W}^\top/\sqrt{d})/\sqrt{d}$ by considering the Stieltjes transform of a related random block matrix, and show that, under least squares regression setting in an asymptotic viewpoint, both the bias and variance have a peak at the interpolation threshold $\psi_2 = 1$ and diverge there when $\lambda \to 0$. Under this framework, according to the randomness stemming from label noise, initialization, and training features, a refined bias-variance decomposition is conducted by [141], [148] and further improved by [143], [149] using the *analysis of variance*. Apart from refined error decomposition schemes, the authors of [140], [142], [144] consider a general setting on convex loss functions, transformation matrix, and activation functions for regression and classification. Here the techniques used for analysis are not limited to RMT. Instead, replica method [150] (a non-rigorous heuristic method from statistical physics) used in [141], [142], [148] and the convex Gaussian min-max (CGMM) theorem [151] used in [144] are two alternative way to derive the desired results. Note that, CGMM requires the data to be Gaussian, which might restrict the application scope of their results but is still a common-used technical tool for max-margin linear classifier [152], boosting classifiers [153], and adversarial training for linear regression [154] in over-parameterized regimes. Admittedly, the applied replica method in statistical physics is quite different from [126] for tackling inverse random matrices in RMT. However, most of the above methods admit the equivalence between the considered data model and the Gaussian covariate model. That means, problem (3) with Gaussian data can be asymptotically equivalent to

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^s} \frac{1}{n} \sum_{i=1}^n \ell\left(y_i, \boldsymbol{\beta}^\top \left(\mu_0 \mathbf{1}_k + \mu_1 \boldsymbol{W}\boldsymbol{x}_i + \mu_\star \boldsymbol{t}_i\right)\right) + \lambda \|\boldsymbol{\beta}\|_2^2,$$

where $\{\boldsymbol{t}_i\}_{i=1}^n \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$, $\mu_0 = \mathbb{E}[\sigma(t)]$, $\mu_1 = \mathbb{E}[t\sigma(t)]$ and $\mu_\star = \mathbb{E}[\sigma(t)^2] - \mu_0^2 - \mu_1^2$ for a standard Gaussian variable t. This equivalence on generalization error in an asymptotic viewpoint is proved in [145].

Different from the above results in an asymptotic view, Jacot et al. [146] present a non-asymptotic result by taking finite-size Stieltjes transform of generalized Wishart matrix, and further argue that random feature models can be close to KRR with an additional regularization. The used technical tool is related to the "calculus of deterministic equivalents" for random matrices [155]. This technique is also used in [139] to derive the exact asymptotic deterministic equivalent of $\mathbb{E}_{\boldsymbol{W}}[(\boldsymbol{Z}\boldsymbol{Z}^\top + n\lambda \boldsymbol{I})^{-1}]$, which captures the asymptotic behavior on double descent. Note that, this work makes no data assumption to match real-world data, which is different from previous work relying on specific data distribution.

7.2 Discussion on Random Features and DNNs

As mentioned, random features models have been fruitfully used to analyze the double descent phenomenon. However, it is non-trivial to transfer results for these models to practical neural networks, which are typically deep but not too wide. There is still a substantial gap between existing theory based on random features and the

modern practice of DNNs in approximation ability. For example, under the spherical data setting, Ghorbani et al. [37] (a more general version in [156] on data distribution) point out that as $n \to \infty$, a random features regression model can only fit the projection of the target function onto the space of degree-ℓ polynomials when $s = \Omega(d^{\ell+1-\delta})$ random features are used for some $\delta > 0$. More importantly, if s, d are taken as large with $s = \Omega(d)$, then the function space by random features can only capture linear functions. Even if we consider the NTK model, it can just capture quadratic functions. That means, both random features and NTK have limited approximation power in the lazy training scheme [35]. In addition, Yehudai and Shamir [157] show that the random features model cannot efficiently approximate a single ReLU neuron as it requires the number of random features to be exponentially large in the feature dimension d. This is consistent with the classical result for kernel approximation in the under-parameterized regime: the random features model, QMC, and quadrature based methods require $s = \Omega(\exp(d))$ to achieve an ϵ approximation error [20].

Admittedly, the above results may appear pessimistic due to the simple architecture. Nevertheless, random features is still an effective tool, at least the first step, for analyzing and understanding DNNs in certain regimes, and we believe its potential has yet to be fully exploited. Note that the random features model is still a strong and universal approximator [158] in the sense that the RKHSs induced by a broad class of random features are dense in the space of continuous functions. While the aforementioned results show that the number of required features may be exponential in the worst case, a more refined analysis can still provide useful insights for DNNs. One potential way forward in deep learning theory is to use the random features model to analyze DNNs *with pruning*. For example, the best paper [159] in the *Seventh International Conference on Learning Representations* (ICLR2019) put forward the following *Lottery Ticket Hypothesis*: a deep neural network with random initialization contains a small sub-network which, when trained in isolation, can compete with the performance of the original one. Malach et al. [160] provide a stronger claim that a randomly-initialized and sufficiently over-parameterized neural network contains a sub-network with nearly the same accuracy as the original one, without any further training. Their analysis points to the equivalence between random features and the sub-network model. As such, the random features model is potentially useful for network pruning [161] in terms of, e.g., guiding the design of neurons pruning for accelerating computations, and understanding network pruning and the full DNNs.

8 Conclusion

In this survey, we systematically review random features based algorithms and their associated theoretical results. We also give an overview on generalization properties of high dimensional random features in over-parameterized regimes on double descent, and discuss the limitations and potential of random features in the theory development for neural networks. Below we provide additional remarks and discuss several open problems that are of interest for future research.

- As a typical data independent method, random features are simpler to implement, easy to parallelize, and naturally apply to streaming or dynamic data. Current efforts on Nyström approximation by a preconditioned gradient solver parallelized with multiple GPUs [162] and quantum algorithms [95] can guide us to design powerful implementation for random features to handling millions/billions data.
- Experimental comparisons show that better kernel approximation does not directly translate to lower generalization errors. We partly answer this question in the current survey but it may be not sufficient to explain this phenomenon. We believe this issue deserves further in-depth study.
- Kernel learning via the spectral density is a popular direction [66], [68], which can be naturally combined with Generative Adversarial Networks (GANs); see [63] for details. In this setting, one may associate the learned model with an implicit probability density that is flexible to characterize the relationships and similarities in the data. This is an interesting area for further research.
- The double descent phenomenon has been observed and studied in random features model by various technical tools under different settings. Current theoretical results, such as those in [126], [139], may be extended to a more general setting with less restricted assumptions on data generation, model formulation, and the target function. Besides, more refined analysis and delicate phenomena beyond double descent have been investigated on the linear model, e.g., multiple descent phenomena [163] and optimal (negative) regularization [164], [165]. Understanding these more delicate phenomena for random features requires further investigation and refined analysis.

We hope that this survey will stimulate further research on the above open problems.

Acknowledgements

The research leading to these results has received funding from the European Research Council under the European Union's Horizon 2020 research and innovation program / ERC Advanced Grant E-DUALITY (787960). This paper reflects only the authors' views and the Union is not liable for any use that may be made of the contained information. This work was supported in part by Research Council KU Leuven: Optimization frameworks for deep kernel machines C14/18/068; Flemish Government: FWO projects: GOA4917N (Deep Restricted Kernel Machines: Methods and Foundations), PhD/Postdoc grant. This research received funding from the Flemish Government (AI Research Program). This work was supported in part by Ford KU Leuven Research Alliance Project KUL0076 (Stability analysis and performance improvement of deep reinforcement learning algorithms), EU H2020 ICT-48 Network TAILOR (Foundations of Trustworthy AI - Integrating Reasoning, Learning and Optimization), Leuven.AI Institute; and in part by the National Natural Science Foundation of China 61977046, in part by National Science Foundation grants CCF-1657420 and CCF-1704828, and in part by SJTU Global Strategic Partnership Fund (2020 SJTU-CORNELL) and Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102).

References

[1] B. Schölkopf and A.J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*, MIT Press, 2003.

[2] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle, *Least Squares Support Vector Machines*, World Scientific, 2002.

[3] M. Kafai and K. Eshghi, "CROification: accurate kernel classification with the efficiency of sparse linear SVM," *IEEE T-PAMI*, vol. 41, no. 1, pp. 34–48, 2019.

[4] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *NeurIPS*, 2007, pp. 1177–1184.

[5] H. Wendland, *Scattered data approximation*, Cambridge university press, 2004.

[6] D. Lopez-Paz, S. Sra, A.J. Smola, Z. Ghahramani, and B. Schölkopf, "Randomized nonlinear component analysis," in *ICML*, 2014, pp. 1359–1367.

[7] Y. Sun, A. Gilbert, and A. Tewari, "But how does it work in theory? Linear SVM with random features," in *NeurIPS*, 2018, pp. 3383–3392.

[8] A. Jacot, F. Gabriel, and C. Hongler, "Neural tangent kernel: Convergence and generalization in neural networks," in *NeurIPS*, 2018, pp. 8571–8580.

[9] S. Arora, S. Du, W. Hu, Z. Li, R. Salakhutdinov, and R. Wang, "On exact computation with an infinitely wide neural net," in *NeurIPS*, 2019, pp. 8139–8148.

[10] A. Zandieh, I. Han, H. Avron, N. Shoham, C. Kim, and J. Shin, "Scaling neural tangent kernels via sketching and random features," *arXiv preprint arXiv:2106.07880*, 2021.

[11] S. Du, K. Hou, B. Póczos, R. Salakhutdinov, R. Wang, and K. Xu, "Graph neural tangent kernel: Fusing graph neural networks with graph kernels," in *NeurIPS*, 2019, pp. 1–11.

[12] D. Zambon, C. Alippi, and L. Livi, "Graph random neural features for distance-preserving graph representations," in *ICML*, 2020, pp. 10968–10977.

[13] K. Choromanski et al., "Rethinking attention with performers," in *ICLR*, 2021.

[14] H. Peng, N. Pappas, D. Yogatama, R. Schwartz, N. Smith, and L. Kong, "Random feature attention," in *ICLR*, 2021, pp. 1–19.

[15] Y. Cao and Q. Gu, "Generalization bounds of stochastic gradient descent for wide and deep neural networks," in *NeurIPS*, 2019, pp. 10835–10845.

[16] S. Arora, S. Du, W. Hu, Z. Li, and R. Wang, "Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks," in *ICML*, 2019, pp. 322–332.

[17] Z. Ji and M. Telgarsky, "Polylogarithmic width suffices for gradient descent to achieve arbitrarily small test error with shallow ReLU networks," in *ICLR*, 2020, pp. 1–8.

[18] F. Yu, A.T. Suresh, K. Choromanski, D. Holtmannrice, and S. Kumar, "Orthogonal random features," in *NeurIPS*, 2016, pp. 1975–1983.

[19] H. Avron, V. Sindhwani, J. Yang, and M. Mahoney, "Quasi-Monte Carlo feature maps for shift-invariant kernels," *JMLR*, vol. 17, no. 1, pp. 4096–4133, 2016.

[20] T. Dao, C. De Sa, and C. Ré, "Gaussian quadrature for kernel features," in *NeurIPS*, 2017, pp. 6107–6117.

[21] M. Munkhoeva, Y. Kapushev, E. Burnaev, and I. Oseledets, "Quadrature-based features for kernel approximation," in *NeurIPS*, 2018, pp. 9147–9156.

[22] A. Saade, F. Caltagirone, I. Carron, L. Daudet, A. Drémeau, S. Gigan, and F. Krzakala, "Random projections through multiple optical scattering: Approximating kernels at the speed of light," in *ICASSP*. IEEE, 2016, pp. 6215–6219.

[23] R. Ohana, J. Wacker, J. Dong, S. Marmin, F. Krzakala, M. Filippone, and L. Daudet, "Kernel computations from large-scale random features obtained by optical processing units," *arXiv preprint arXiv:1910.09880*, 2019.

[24] D.J. Sutherland and J. Schneider, "On the error of random Fourier features," in *UAI*, 2015, pp. 862–871.

[25] Z. Li, J. Ton, D. Oglic, and D. Sejdinovic, "Towards a unified analysis of random Fourier features," in *ICML*, 2019, pp. 3905–3914.

[26] I. J. Schoenberg, "Positive definite functions on spheres," *Duke Math. J.*, vol. 9, no. 1, pp. 96–108, 1942.

[27] Alex J. Smola, Zoltan L. Ovari, and Robert C. Williamson, "Regularization with dot-product kernels," in *NeurIPS*, 2001, pp. 308–314.

[28] C. Müller, *Spherical harmonics*, vol. 17, Springer, 2006.

[29] G. Huang, Q. Zhu, and C. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, 2006.

[30] Y. Cho and L. Saul, "Kernel methods for deep learning," in *NeurIPS*, 2009, pp. 342–350.

[31] C. Williams, "Computing with infinite networks," in *NeurIPS*, 1997, pp. 295–301.

[32] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," *arXiv:1606.08415*, 2016.

[33] A. Daniely, R. Frostig, and Y. Singer, "Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity," in *NeurIPS*, 2016, pp. 2253–2261.

[34] J. Lee, Y. Bahri, R. Novak, S. Schoenholz, J. Pennington, and J. Sohl-Dickstein, "Deep neural networks as Gaussian Processes," in *ICLR*, 2018.

[35] L. Chizat, E. Oyallon, and F. Bach, "On lazy training in differentiable programming," in *NeurIPS*, 2019, pp. 2933–2943.

[36] A. Bietti and J. Mairal, "On the inductive bias of neural tangent kernels," in *NeurIPS*, 2019, pp. 12873–12884.

[37] B. Ghorbani, S. Mei, T. Misiakiewicz, and A. Montanari, "Linearized two-layers neural networks in high dimension," *Ann. Stat.*, 2019.

[38] A. Bietti and F. Bach, "Deep equals shallow for ReLU networks in kernel regimes," in *ICLR*, 2021.

[39] P. Kar and H. Karnick, "Random feature maps for dot product kernels," in *AISTATS*, 2012, pp. 583–591.

[40] N. Pham and R. Pagh, "Fast and scalable polynomial kernels via explicit feature maps," in *KDD*, 2013, pp. 239–247.

[41] M. Meister, T. Sarlos, and D. Woodruff, "Tight dimensionality reduction for sketching low degree polynomial kernels," in *NeurIPS*, 2019, pp. 9475–9486.

[42] H. Avron, H. Nguyen, and D. Woodruff, "Subspace embeddings for the polynomial kernel," in *NeurIPS*, 2014, pp. 2258–2266.

[43] D. Woodruff and A. Zandieh, "Near input sparsity time kernel embeddings via adaptive sampling," in *ICML*, 2020, pp. 10324–10333.

[44] J. Pennington, F. Yu, and S. Kumar, "Spherical random features for polynomial kernels," in *NeurIPS*, 2015, pp. 1846–1854.

[45] F. Liu, X. Huang, L. Shi, J. Yang, and J.A.K. Suykens, "A double-variational Bayesian framework in random Fourier features for indefinite kernels," *IEEE T-NNLS*, vol. 31, no. 8, pp. 2965–2979, 2020.

[46] F. Liu, X. Huang, Y. Chen, and J.A.K. Suykens, "Fast learning in reproducing kernel Kreĭn spaces via signed measures," in *AISTATS*, 2021, pp. 1–11.

[47] Q. Le, T. Sarlós, and A.J. Smola, "FastFood—approximating kernel expansions in loglinear time," in *ICML*, 2013, pp. 244–252.

[48] K. Choromanski and V. Sindhwani, "Recycling randomness with structure for sublinear time kernel expansions," in *ICML*, 2016, pp. 2502–2510.

[49] C. Feng, Q. Hu, and S. Liao, "Random feature mapping with signed circulant matrix projection," in *IJCAI*, 2015.

[50] Ping Li, "Linearized GMM kernels and normalized random Fourier features," in *ACM SIGKDD*, 2017, pp. 315–324.

[51] K.M. Choromanski, M. Rowland, and A. Weller, "The unreasonable effectiveness of structured random orthogonal embeddings," in *NeurIPS*, 2017, pp. 219–228.

[52] J. Yang, V. Sindhwani, H. Avron, and M. Mahoney, "Quasi-Monte Carlo feature maps for shift-invariant kernels," in *ICML*, 2014, pp. 485–493.

[53] Y. Lyu, "Spherical structured feature maps for kernel approximation," in *ICML*, 2017, pp. 2256–2264.

[54] W. Shen, Z. Yang, and J. Wang, "Random features for shift-invariant kernels with moment matching," in *AAAI*, 2017, pp. 2520–2526.

[55] A. Rudi, D. Calandriello, L. Carratino, and L. Rosasco, "On fast leverage score sampling and optimal learning," in *NeurIPS*, 2018, pp. 5672–5682.

[56] F. Liu, X. Huang, Y. Chen, J. Yang, and J.A.K. Suykens, "Random Fourier features via fast surrogate leverage weighted sampling," in *AAAI*, 2020, pp. 4844–4851.

[57] T. Erdélyi, C. Musco, and C. Musco, "Fourier sparse leverage scores and approximate kernel learning," in *NeurIPS*, 2020.

[58] A. Rahimi and B. Recht, "Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning," in *NeurIPS*, 2009, pp. 1313–1320.

[59] W. Chang, C. Li, Y. Yang, and B. Póczos, "Data-driven random Fourier features using Stein effect," in *IJCAI*, 2017, pp. 1497–1503.

[60] A. Sinha and J.C. Duchi, "Learning kernels with random features," in *NeurIPS*, 2016, pp. 1298–1306.

[61] S. Shahrampour, A. Beirami, and V. Tarokh, "On data-dependent random features for improved generalization in supervised learning," in *AAAI*, 2018, pp. 4026–4033.

[62] R. Agrawal, T. Campbell, J. Huggins, and T. Broderick, "Data-dependent compression of random features for large-scale kernel approximation," in *AISTATS*, 2019, pp. 1822–1831.

[63] C. Li, W. Chang, Y. Mroueh, Y. Yang, and B. Poczos, "Implicit kernel learning," in *AISTATS*, 2019, pp. 2007–2016.

[64] F. Yu, S. Kumar, H. Rowley, and S. Chang, "Compact nonlinear maps and circulant extensions," *arXiv preprint arXiv:1503.03893*, 2015.

[65] B. Bullins, C. Zhang, and Y. Zhang, "Not-so-random features," in *ICLR*, 2018.

[66] A. Wilson and R. Adams, "Gaussian process kernels for pattern discovery and extrapolation," in *ICML*, 2013, pp. 1067–1075.

[67] Z. Yang, A. Wilson, A.J. Smola, and L. Song, "À la carte–learning fast kernels," in *AISTATS*, 2015, pp. 1098–1106.

[68] Z. Shen, M. Heinonen, and S. Kaski, "Harmonizable mixture kernels with variational Fourier features," in *AISTATS*. PMLR, 2019.

[69] J. Oliva, A. Dubey, A. Wilson, B. Póczos, J. Schneider, and E. Xing, "Bayesian nonparametric kernel learning," in *AISTATS*, 2016, pp. 1078–1086.

[70] J. Zhang, A. May, T. Dao, and C. Re, "Low-precision random Fourier features for memory-constrained kernel approximation," in *AISTATS*, 2019, pp. 1264–1274.

[71] B. Dai, B. Xie, N. He, Y. Liang, A. Raj, M. Balcan, and L. Song, "Scalable kernel methods via doubly stochastic gradients," in *NeurIPS*, 2014, pp. 3041–3049.

[72] M. Bojarski, A. Choromanska, K. Choromanski, F. Fagan, C. Gouy-Pailler, A. Morvan, N. Sakr, T. Sarlos, and J. Atif, "Structured adaptive and random spinners for fast machine learning computations," in *AISTATS*, 2017, pp. 1020–1029.

[73] H. Niederreiter, *Random number generation and quasi-Monte Carlo methods*, vol. 63, SIAM, 1992.

[74] F. Bach, "On the equivalence between kernel quadrature rules and random feature expansions," *JMLR*, vol. 18, no. 1, pp. 714–751, 2017.

[75] H. Avron, M. Kapralov, C. Musco, C. Musco, A. Velingker, and A. Zandieh, "Random Fourier features for kernel ridge regression: Approximation bounds and statistical guarantees," in *ICML*, 2017, pp. 253–262.

[76] Xiaoyun Li and Ping Li, "Quantization algorithms for random fourier features," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2021, pp. 6369–6380.

[77] T. Yang, Y. Li, M. Mahdavi, R. Jin, and Z. Zhou, "Nyström method vs random Fourier features: a theoretical and empirical comparison," in *NeurIPS*, 2012, pp. 476–484.

[78] B. Xie, Y. Liang, and L. Song, "Scale up nonlinear component analysis with doubly stochastic gradients," in *NeurIPS*, 2015, pp. 2341–2349.

[79] X. Li, B. Gu, S. Ao, H. Wang, and C. Ling, "Triply stochastic gradients on multiple kernel learning.," in *AISTATS*, 2017, pp. 1–9.

[80] K. Choromanski, M. Rowland, W. Chen, and A. Weller, "Unifying orthogonal Monte Carlo methods," in *ICML*, 2019, pp. 1203–1212.

[81] K. Choromanski, M. Rowland, T. Sarlós, V. Sindhwani, R. Turner, and A. Weller, "The geometry of random features," in *AISTATS*, 2018, pp. 1–9.

[82] J.S. Brauchart and P.J. Grabner, "Distributing many points on spheres: minimal energy and designs," *J. of Complex.*, vol. 31, no. 3, pp. 293–326, 2015.

[83] Y. LYU, Y. Yuan, and I. Tsang, "Subgroup-based rank-1 lattice quasi-monte carlo," in *NeurIPS*, 2020.

[84] G. Evans, *Practical numerical integration*, Wiley New York, 1993.

[85] A. Genz and J. Monahan, "Stochastic integration rules for infinite regions," *SIAM J. on Sci. Comput.*, vol. 19, no. 2, pp. 426–439, 1998.

[86] A. Genz and J. Monahan, "A stochastic algorithm for high-dimensional integrals over unbounded regions with gaussian weight," *J. of Comput. and Appli. Math.*, vol. 112, no. 1-2, pp. 71–81, 1999.

[87] F. Heiss and V. Winschel, "Likelihood approximation by numerical integration on sparse grids," *J. of Econometrics*, vol. 144, no. 1, pp. 62–80, 2008.

[88] A. Belhadji, R. Bardenet, and P. Chainais, "Kernel quadrature with dpps," in *NeurIPS*, 2019, pp. 1–11.

[89] F. Liu, X. Huang, Y. Chen, and J.A.K. Suykens, "Towards a unified quadrature framework for large-scale kernel machines," *arXiv:2011.01668*, 2020.

[90] F. Briol, C.J. Oates, J. Cockayne, W.Y. Chen, and M. Girolami, "On the sampling problem for kernel quadrature," in *ICML*, 2017, pp. 586–595.

[91] B. Gauthier and J.A.K. Suykens, "Optimal quadrature-sparsification for integral operator approximation," *SIAM J. Sci. Comput.*, vol. 40, no. 5, pp. A3636–A3674, 2018.

[92] Y. Wang and S. Shahrampour, "A general scoring rule for randomized kernel approximation with application to canonical correlation analysis," *arXiv preprint arXiv:1910.05384*, 2019.

[93] Tong Zhang, "Learning bounds for kernel regression using effective data dimensionality," *Neural Comput.*, vol. 17, no. 9, pp. 2077–2098, 2005.

[94] F. Bach, "Sharp analysis of low-rank kernel matrix approximations," in *COLT*, 2013, pp. 185–209.

[95] H. Yamasaki, S. Subramanian, S. Sonoda, and M. Koashi, "Fast quantum algorithm for learning with optimized random features," in *NeurIPS*, 2020, pp. 1–10.

[96] A. Alaoui and M. Mahoney, "Fast randomized kernel ridge regression with statistical guarantees," in *NeurIPS*, 2015, pp. 775–783.

[97] D. Calandriello, A. Lazaric, and M. Valko, "Distributed adaptive sampling for kernel matrix approximation," in *AISTATS*, 2017, pp. 1421–1429.

[98] B.K. Sriperumbudur and Z. Szabó, "Optimal rates for random Fourier features," in *NeurIPS*, 2015, pp. 1144–1152.

[99] J. Honorio and Y. Li, "The error probability of random Fourier features is dimensionality independent," *arXiv preprint arXiv:1710.09953*, 2017.

[100] A. Rudi and L. Rosasco, "Generalization properties of learning with random features," in *NeurIPS*, 2017, pp. 3215–3225.

[101] A. Rahimi and B. Recht, "Uniform approximation of functions with random bases," in *Proc. of the Conf. on Commu., Control, and Comput.* IEEE, 2008, pp. 555–561.

[102] Danica J. Sutherland and Jeff Schneider, "On the error of random Fourier features," in *Conference on Uncertainty in Artificial Intelligence*, 2015, pp. 862–871.

[103] Haim Avron, Michael Kapralov, Cameron Musco, Christopher Musco, Ameya Velingker, and Amir Zandieh, "Random Fourier features for kernel ridge regression: Approximation bounds and statistical guarantees," in *International Conference on Machine Learning*, 2017, pp. 253–262.

[104] M. Ghashami, D. Perry, and J. Phillips, "Streaming kernel principal component analysis," in *AISTATS*, 2016, pp. 1365–1374.

[105] B. Sriperumbudur and N. Sterge, "Statistical consistency of kernel PCA with random features," *arXiv:1706.06296*, 2017.

[106] E. Ullah, P. Mianjy, T.V. Marinov, and R. Arora, "Streaming kernel PCA with $\tilde{O}(\sqrt{n})$ random features," in *NeurIPS*, 2018, pp. 7311–7321.

[107] F. Cucker and D. Zhou, *Learning theory: an approximation theory viewpoint*, vol. 24, Cambridge University Press, 2007.

[108] I. Steinwart and C. Andreas, *Support Vector Machines*, Springer Science and Business Media, 2008.

[109] G. Blanchard and N. Krämer, "Optimal learning rates for kernel conjugate gradient regression," in *NeurIPS*, 2010, pp. 226–234.

[110] A. Caponnetto and E. De Vito, "Optimal rates for the regularized least-squares algorithm," *Found. Comput. Math.*, vol. 7, no. 3, pp. 331–368, 2007.

[111] J. Shawe-Taylor, C. Williams, N. Cristianini, and J. Kandola, "On the eigenspectrum of the gram matrix and its relationship to the operator eigenspectrum," in *COLT*. Springer, 2002, pp. 23–40.

[112] S. Smale and D.X. Zhou, "Learning theory estimates via integral operators and their approximations," *Constructive Approx.*, vol. 26, no. 2, pp. 153–172, 2007.

[113] Z. Guo and L. Shi, "Optimal rates for coefficient-based regularized regression," *ACHA*, vol. 47, no. 3, pp. 662–701, 2019.

[114] S. Lin, X. Guo, and D. Zhou, "Distributed learning with regularized least squares," *JMLR*, vol. 18, no. 1, pp. 3202–3232, 2017.

[115] V. Koltchinskii, *Oracle Inequalities in Empirical Risk Minimization and Sparse Recovery Problems*, vol. 2033, Springer Science & Business Media, 2011.

[116] Peter L Bartlett, Olivier Bousquet, and Shahar Mendelson, "Local rademacher complexities," *Annals of Statistics*, vol. 33, no. 4, pp. 1497–1537, 2005.

[117] Z. Li, J.F. Ton, D. Oglic, and D. Sejdinovic, "Towards a unified analysis of random fourier features," *JMLR*, vol. 22, no. 108, pp. 1–51, 2021.

[118] L. Carratino, A. Rudi, and L. Rosasco, "Learning with SGD and random features," in *NeurIPS*, 2018, pp. 10212–10223.

[119] S. Wang, "Simple and almost assumption-free out-of-sample bound for random feature mapping," *arXiv preprint arXiv:1909.11207*, 2019.

[120] R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin, "LIBLINEAR: a library for large linear classification," *JMLR*, vol. 9, pp. 1871–1874, 2008.

[121] G. Loosli, S. Canu, and L. Bottou, "Training invariant support vector machines using selective sampling," *Large Scale Kernel Mach.*, vol. 2, 2007.

[122] S. Arora, S. Du, Z. Li, R. Salakhutdinov, R. Wang, and D. Yu, "Harnessing the power of infinitely wide deep nets on small-data tasks," in *ICLR*, 2020.

[123] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015, pp. 448–456.

[124] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009, pp. 248–255.

[125] T. Hastie, A. Montanari, S. Rosset, and R. Tibshirani, "Surprises in high-dimensional ridgeless least squares interpolation," *arXiv preprint arXiv:1903.08560*, 2019.

[126] S. Mei and A. Montanari, "The generalization error of random features regression: Precise asymptotics and double descent curve," *arXiv preprint arXiv:1908.05355*, 2019.

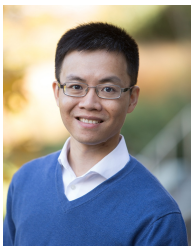
[127] T. Liang, A. Rakhlin, and X. Zhai, "On the multiple descent of minimum-norm interpolants and restricted lower isometry of kernels," in *COLT*, 2019, pp. 1–32.

[128] F. Liu, Z. Liao, and J.A.K. Suykens, "Kernel regression in high dimensions: Refined analysis beyond double descent," in *AISTATS*, 2021, pp. 1–11.

[129] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," *arXiv preprint arXiv:1611.03530*, 2016.
[130] M. Belkin, D. Hsu, S. Ma, and S. Mandal, "Reconciling modern machine-learning practice and the classical bias–variance trade-off," *PNAS*, vol. 116, no. 32, pp. 15849–15854, 2019.
[131] P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, and I. Sutskever, "Deep double descent: Where bigger models and more data hurt," in *ICLR*, 2019.
[132] M. Belkin, S. Ma, and S. Mandal, "To understand deep learning we need to understand kernel learning," in *ICML*, 2018, pp. 541–549.
[133] Yann Lecun, Leon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*.
[134] Felipe Cucker and Steve Smale, "On the mathematical foundations of learning," *Bulletin of the American mathematical society*, vol. 39, no. 1, pp. 1–49, 2002.
[135] Z. Allen-Zhu, Y. Li, and Y. Liang, "Learning and generalization in overparameterized neural networks, going beyond two layers," in *NeurIPS*, 2019, pp. 6158–6169.
[136] T. Tao, *Topics in random matrix theory*, American Mathematical Society, 2012.
[137] J. Pennington and P. Worah, "Nonlinear random matrix theory for deep learning," in *NeurIPS*, 2017, pp. 2634–2643.
[138] Z. Liao and R. Couillet, "On the spectrum of random features maps of high dimensional data," in *ICML*, 2018, pp. 3063–3071.
[139] Z. Liao, R. Couillet, and M. Mahoney, "A random matrix analysis of random fourier features: beyond the gaussian kernel, a precise phase transition, and the corresponding double descent," in *NeurIPS*, 2020.
[140] J. Ba, M. Erdogdu, T. Suzuki, D. Wu, and T. Zhang, "Generalization of two-layer neural networks: an asymptotic viewpoint," in *ICLR*, 2020, pp. 1–8.
[141] S. d'Ascoli, M. Refinetti, G. Biroli, and F. Krzakala, "Double trouble in double descent: Bias and variance(s) in the lazy regime," *arXiv preprint arXiv:2003.01054*, 2020.
[142] F. Gerace, B. Loureiro, F. Krzakala, M. Mézard, and L. Zdeborová, "Generalisation error in learning with random features and the hidden manifold model," in *International Conference on Machine Learning*, 2020, pp. 3452–3462.
[143] B. Adlam and J. Pennington, "Understanding double descent requires a fine-grained bias-variance decomposition," in *NeurIPS*, 2020.
[144] O. Dhifallah and Y.M. Lu, "A precise performance analysis of learning with random features," *arXiv:2008.11904*, 2020.
[145] H. Hu and Y.M. Lu, "Universality laws for high-dimensional learning with random features," *arXiv:2009.07669*, 2020.
[146] A. et al. Jacot, "Implicit regularization of random feature models," in *ICML*, 2020.
[147] M. Belkin, D. Hsu, and J. Xu, "Two models of double descent for weak features," *SIAM J. Math. Data Sci.*, vol. 2, no. 4, pp. 1167–1180, 2020.
[148] J.W. Rocks and P. Mehta, "Memorizing without overfitting: Bias, variance, and interpolation in over-parameterized models," *arXiv:2010.13933*, 2020.
[149] L. Lin and E. Dobriban, "What causes the test error? going beyond bias-variance via anova," *arXiv:2010.05170*, 2020.
[150] M. Mézard, G. Parisi, and M.A. Virasoro, *Spin glass theory and beyond: An Introduction to the Replica Method and Its Applications*, vol. 9, World Scientific Publishing Company, 1987.
[151] C. Thrampoulidis, S. Oymak, and B. Hassibi, "Regularized linear regression: A precise analysis of the estimation error," in *COLT*, 2015, pp. 1683–1709.
[152] A. Montanari, F. Ruan, Y. Sohn, and J. Yan, "The generalization error of max-margin linear classifiers: High-dimensional asymptotics in the overparametrized regime," *arXiv preprint arXiv:1911.01544*, 2019.
[153] T. Liang and P. Sur, "A precise high-dimensional asymptotic theory for boosting and min-ℓ_1-norm interpolated classifiers," *arXiv:2002.01586*, 2020.
[154] A. Javanmard, M. Soltanolkotabi, and H. Hassani, "Precise tradeoffs in adversarial training for linear regression," in *COLT*, 2020, pp. 2034–2078.
[155] C. Louart, Z. Liao, and R. Couillet, "A random matrix approach to neural networks," *Ann. Appli. Prob.*, vol. 28, no. 2, pp. 1190–1248, 2018.
[156] S. Mei, T. Misiakiewicz, and A. Montanari, "Generalization error of random features and kernel methods: hypercontractivity and kernel matrix concentration," *arXiv:2101.10588*, 2021.
[157] G. Yehudai and O. Shamir, "On the power and limitations of random features for understanding neural networks," in *NeurIPS*, 2019, pp. 6594–6604.
[158] Y. Sun, A. Gilbert, and A. Tewari, "On the approximation properties of random ReLU features," *arXiv preprint arXiv:1810.04374*, 2018.
[159] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks," in *ICLR*, 2019.
[160] E. Malach, G. Yehudai, S. Shalev-Shwartz, and O. Shamir, "Proving the lottery ticket hypothesis: Pruning is all you need," *arXiv preprint arXiv:2002.00585*, 2020.
[161] H. Hu, R. Peng, Y. Tai, and C. Tang, "Network trimming: A data-driven neuron pruning approach towards efficient deep architectures," *arXiv preprint arXiv:1607.03250*, 2016.
[162] G. Meanti, L. Carratino, L. Rosasco, and A. Rudi, "Kernel methods through the roof: Handling billions of points efficiently," in *NeurIPS*, 2020.
[163] L. Chen, Y. Min, M. Belkin, and A. Karbasi, "Multiple descent: Design your own generalization curve," *arXiv:2008.01036*, 2020.
[164] Denny Wu and Ji Xu, "On the optimal weighted ℓ_2 regularization in overparameterized linear regression," in *NeurIPS*, 2020, pp. 1–11.
[165] D. Kobak, J. Lomond, and B. Sanchez, "The optimal ridge penalty for real-world high-dimensional data can be zero or negative due to the implicit ridge regularization," *JMLR*, vol. 21, no. 169, pp. 1–16, 2020.
[166] B. Ghorbani, S. Mei, T. Misiakiewicz, and A. Montanari, "Limitations of lazy training of two-layers neural network," in *NeurIPS*, 2019, pp. 9108–9118.
[167] F. Li, C. Ionescu, and C. Sminchisescu, "Random Fourier approximations for skewed multiplicative histogram kernels," in *Proc. of Joint Pattern Recog. Symp.* Springer, 2010, pp. 262–271.
[168] A. Vedaldi and A. Zisserman, "Efficient additive kernels via explicit feature maps," *IEEE T-PAMI*, vol. 34, no. 3, pp. 480–492, 2012.
[169] C. Cortes, M. Mohri, and A. Rostamizadeh, "Two-stage learning kernel algorithms," in *ICML*, 2010, pp. 239–246.
[170] T. Campbell and T. Broderick, "Bayesian coreset construction via greedy iterative geodesic ascent," in *ICML*, 2018, pp. 698–706.
[171] T. Campbell and T. Broderick, "Automated scalable Bayesian inference via Hilbert coresets," *JMLR*, vol. 20, no. 1, pp. 551–588, 2019.
[172] R. Hamid, Y. Xiao, A. Gittens, and D. Decoste, "Compact random feature maps," in *ICML*, 2014, pp. 19–27.
[173] S. Remes, M. Heinonen, and S. Kaski, "Non-stationary spectral kernels," in *NeurIPS*, 2017, pp. 4642–4651.
[174] J.F. Ton, S. Flaxman, D. Sejdinovic, and S. Bhatt, "Spatial mapping with Gaussian processes and nonstationary Fourier features," *Spatial Stat.*, vol. 28, pp. 59–78, 2018.
[175] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
[176] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," *Tech. report, Univ. Toronto*, 2009.

Fanghui Liu (M'19-) received the B.E. degree in Automation from Harbin Institute of Technology, China, and the Ph.D. degree from Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, China, in 2014 and 2019, respectively. He is currently a postdoctoral researcher in ESAT-STADIUS, KU Leuven, Belgium. His research areas mainly include machine learning, kernel methods, and learning theory.

Xiaolin Huang (S'10-M'12-SM'18) received the B.S. degree in control science and engineering, and the B.S. degree in applied mathematics from Xi'an Jiaotong University, Xi'an, China in 2006. In 2012, he received the Ph.D. degree in control science and engineering from Tsinghua University, Beijing, China. From 2012 to 2015, he worked as a postdoctoral researcher in ESAT-STADIUS, KU Leuven, Leuven, Belgium. After that he was selected as an Alexander von Humboldt Fellow and working in Pattern Recognition Lab, the Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany, where he was appointed as a group head. From 2016, he has been an Associate Professor at Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai, China. In 2017, he has been awarded as "1000-Talent"(Young Program). His current research areas include machine learning, optimization, and their applications.



Yudong Chen is an Assistant Professor with the School of Operations Research and Information Engineering at Cornell University. He obtained his Ph.D. degree in Electrical and Computer Engineering in 2013 from The University of Texas at Austin, and M.S. and B.S. degrees in Control Science and Engineering from Tsinghua University. He was a postdoctoral scholar in the Electrical Engineering and Computer Sciences department at the University of California, Berkeley from 2013 to 2015. He has served on the senior program committees of AAAI and AISTATS. His research work lies in machine learning, reinforcement learning, high-dimensional statistics, and optimization, with applications in network scheduling, wireless communication, and financial systems.

Johan A. K. Suykens (SM'05-F'15) was born in Willebroek Belgium, May 18 1966. He received the master degree in Electro-Mechanical Engineering and the PhD degree in Applied Sciences from the Katholieke Universiteit Leuven, in 1989 and 1995, respectively. In 1996 he has been a Visiting Postdoctoral Researcher at the University of California, Berkeley. He has been a Postdoctoral Researcher with the Fund for Scientific Research FWO Flanders and is currently a full Professor with KU Leuven.

He is author of the books "Artificial Neural Networks for Modelling and Control of Non-linear Systems" (Kluwer Academic Publishers) and "Least Squares Support Vector Machines" (World Scientific), co-author of the book "Cellular Neural Networks, Multi-Scroll Chaos and Synchronization" (World Scientific) and editor of the books "Nonlinear Modeling: Advanced Black-Box Techniques" (Kluwer Academic Publishers), "Advances in Learning Theory: Methods, Models and Applications" (IOS Press) and "Regularization, Optimization, Kernels, and Support Vector Machines" (Chapman & Hall/CRC). In 1998 he organized an International Workshop on Nonlinear Modelling with Time-series Prediction Competition. He has served as associate editor for the IEEE Transactions on Circuits and Systems (1997-1999 and 2004-2007), the IEEE Transactions on Neural Networks (1998-2009), the IEEE Transactions on Neural Networks and Learning Systems (from 2017) and the IEEE Transactions on Artificial Intelligence (from April 2020). He received an IEEE Signal Processing Society 1999 Best Paper Award, a 2019 Entropy Best Paper Award and several Best Paper Awards at International Conferences. He is a recipient of the International Neural Networks Society INNS 2000 Young Investigator Award for significant contributions in the field of neural networks. He has served as a Director and Organizer of the NATO Advanced Study Institute on Learning Theory and Practice (Leuven 2002), as a program co-chair for the International Joint Conference on Neural Networks 2004 and the International Symposium on Nonlinear Theory and its Applications 2005, as an organizer of the International Symposium on Synchronization in Complex Networks 2007, a co-organizer of the NIPS 2010 workshop on Tensors, Kernels and Machine Learning, and chair of ROKS 2013.He has been awarded an ERC Advanced Grant 2011 and 2017, has been elevated IEEE Fellow 2015 for developing least squares support vector machines, and is ELLIS Fellow. He is currently serving as program director of Master AI at KU Leuven.