RANDOM MATRICES IN SERVICE OF ML FOOTPRINT: TERNARY RANDOM FEATURES WITH NO PERFORMANCE LOSS

Anonymous authors
Paper under double-blind review

ABSTRACT

In this article, we investigate the spectral behavior of random features kernel matrices of the type $\mathbf{K} = \mathbb{E}_{\mathbf{w}} \left[\sigma \left(\mathbf{w}^\mathsf{T} \mathbf{x}_i \right) \sigma \left(\mathbf{w}^\mathsf{T} \mathbf{x}_j \right) \right]_{i,j=1}^n$, with nonlinear function $\sigma(\cdot)$, data $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^p$, and random projection vector $\mathbf{w} \in \mathbb{R}^p$ having i.i.d. entries. In a high-dimensional setting where the number of data n and their dimension p are both large and comparable, we show, under a Gaussian mixture model for the data, that the eigenspectrum of $\mathbf{K}$ is *independent* of the distribution of the i.i.d. (zero-mean and unit-variance) entries of $\mathbf{w}$, and *only* depends on $\sigma(\cdot)$ via its (generalized) Gaussian moments $\mathbb{E}_{z \sim \mathcal{N}(0,1)}[\sigma'(z)]$ and $\mathbb{E}_{z \sim \mathcal{N}(0,1)}[\sigma''(z)]$. As a result, for any kernel matrix $\mathbf{K}$ of the form above, we propose a novel random features technique, called Ternary Random Features (TRFs), that (i) asymptotically yields the same limiting kernel as the original $\mathbf{K}$ in a spectral sense and (ii) can be computed and stored much more efficiently, by wisely tuning (in a *data-dependent* manner) the function σ and the random vector $\mathbf{w}$, both taking values in $\{-1, 0, 1\}$. The computation of the proposed random features requires no multiplication, and a factor of b times less bits for storage compared to classical random features such as random Fourier features, with b the number of bits to store full precision values. Besides, it appears in our experiments on real data that the substantial gains in computation and storage are accompanied with somewhat improved performances compared to state-of-the-art random features methods.

1 INTRODUCTION

Kernel methods are among the most powerful machine learning approaches with a wide range of successful applications (Schölkopf & Smola, 2018) which, however, suffer from scalability issues in large-scale problems, due to their high space and time complexities (with respect to the number of data n). To address this key limitation, a myriad of random features based kernel approximation techniques have been proposed (Rahimi & Recht, 2008; Liu et al., 2020): random features methods randomly project the data to obtain low-dimensional nonlinear representations that approximate the original kernel features. This allows practitioners to apply them with a large saving in both time and space, to various kernel-based downstream tasks such as kernel spectral clustering (Von Luxburg, 2007), kernel principal component analysis (Schölkopf et al., 1997), kernel canonical correlation analysis (Lai & Fyfe, 2000), kernel ridge regression (Vovk, 2013), to name a few. A wide variety of these kernels can be written, for data $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^p$, under the form

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \mathbb{E}_{\mathbf{w}} \left[\sigma \left(\mathbf{w}^\mathsf{T} \mathbf{x}_i \right) \sigma \left(\mathbf{w}^\mathsf{T} \mathbf{x}_j \right) \right] \tag{1}$$

with $\mathbf{w} \in \mathbb{R}^p$ having i.i.d. entries, which can be "well approximated" by a sample mean $\frac{1}{m} \sum_{t=1}^m \sigma \left(\mathbf{w}_t^\mathsf{T} \mathbf{x}_i \right) \sigma \left(\mathbf{w}_t^\mathsf{T} \mathbf{x}_j \right)$ over m random features for m sufficiently large. For instance, taking $\sigma(x) = [\cos(x), \ \sin(x)]$ and $\mathbf{w}$ with i.i.d. standard Gaussian entries, one obtains the popular Random Fourier Features (RFFs) that approximate the Gaussian kernel (and the Laplacian kernel for Cauchy distributed $\mathbf{w}$ with the same choice of σ) (Rahimi & Recht, 2008); for $\sigma(x) = \max(x, 0)$, one approximates the first order Arc-cosine kernel; and the zeroth order Arc-cosine kernel (Cho, 2012) with $\sigma(x) = (1 + \operatorname{sign}(x))/2$, etc.

As shall be seen subsequently, (random) neural networks are, to a large extent, connected to *kernel matrices* of the form (1). More specifically, the classification or regression performance at the output of random neural networks are functionals of random matrices that fall into the wide class of kernel random matrices. Perhaps more surprisingly, this connection still exists for *deep neural networks* which are (i) randomly initialized and (ii) trained with gradient descent, as testified by the recent works on *neural tangent kernels* (Jacot et al., 2018), by considering the "infinitely many neurons" limit, that is, the limit where the network widths of all layers go to infinity simultaneously. This close connection between neural networks and kernels has triggered a renewed interest for the theoretical investigation of deep neural networks from various perspectives, including optimization (Du et al., 2019; Chizat et al., 2019), generalization (Allen-Zhu et al., 2018; Arora et al., 2019; Bietti & Mairal, 2019), and learning dynamics (Lee et al., 2019; Advani et al., 2020; Liao & Couillet, 2018a). These works shed new light on the theoretical understanding of deep neural network models and specifically demonstrate the significance of studying networks with random weights and their associated kernels to assess the mechanisms underlying more elaborate deep networks.

In this article, we consider the random features kernel of the type (1), which can also be seen as the limiting kernel of a single-hidden-layer neural network with a random first layer. By assuming a high-dimensional Gaussian Mixture Model (GMM) for the data $\{\mathbf{x}_i\}_{i=1}^n$ with $\mathbf{x}_i \in \mathbb{R}^p$, we show that the *centered* kernel matrix[1]

$$\mathbf{K} \triangleq \mathbf{P}\{\kappa(\mathbf{x}_i, \mathbf{x}_j)\}_{i,j=1}^n \mathbf{P}, \quad \mathbf{P} \triangleq \mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\mathsf{T}, \tag{2}$$

is asymptotically (as $n, p \to \infty$ with $p/n \to c \in (0, \infty)$) equivalent, in a spectral sense, to another kernel matrix $\tilde{\mathbf{K}}$ which depends on the GMM data statistics and the generalized Gaussian moments $\mathbb{E}[\sigma'(z)], \mathbb{E}[\sigma''(z)]$ of the activation function $\sigma(\cdot)$, but is *independent* of the specific law of the i.i.d. entries of the random vector $\mathbf{w}$, as long as they are of zero mean and unit variance. As such, one can design novel random features schemes with limiting kernels asymptotically equivalent to the original $\mathbf{K}$. For instance, define

$$\kappa^{ter}(\mathbf{x}_i, \mathbf{x}_j) \triangleq \mathbb{E}_{\mathbf{w}^{ter}}\left[\sigma^{ter}\left(\mathbf{x}_i^\mathsf{T}\mathbf{w}^{ter}\right)\sigma^{ter}\left(\mathbf{x}_j^\mathsf{T}\mathbf{w}^{ter}\right)\right] \tag{3}$$

with $\mathbf{w}^{ter} \in \mathbb{R}^p$ having i.i.d. entries taking value $w_i^{ter} = 0$ (with probability ϵ) and value

$$w_i^{ter} \in \left\{-(1-\epsilon)^{-\frac{1}{2}}, (1-\epsilon)^{-\frac{1}{2}}\right\} \tag{4}$$

each with probability $1/2 - \epsilon/2$, where $\epsilon \in [0, 1)$ represents the *level of sparsity* of $\mathbf{w}$, and

$$\sigma^{ter}(t) = -1 \cdot \delta_{t<s_-} + 1 \cdot \delta_{t>s_+} \tag{5}$$

for some thresholds $s_- < s_+$, which can be chosen to match the generalized Gaussian moments $\mathbb{E}[\sigma'(z)], \mathbb{E}[\sigma''(z)]$ of *any* σ function (e.g., ReLU, cos, sin) widely used in random features or neural network contexts. The proposed Ternary Random Features (TRFs, with limiting kernel κ^{ter} that asymptotically matches *any* random features kernel) has the computational advantage of being sparse and not requiring multiplications but only additions, as well as the storage advantage of being only composed of a finite set of words, e.g., $\{-1, 0, 1\}$ for $\epsilon = 0$.

Given the urgent need for environmentally-friendly but still efficient neural networks such as binary neural networks (Hubara et al., 2016; Lin et al., 2015; Zhu et al., 2016; Qin et al., 2020; Hubara et al., 2016), pruned neural networks (Liu et al., 2015; Han et al., 2015a;b), weights-quantized neural networks (Gupta et al., 2015; Gong et al., 2014), we believe that our analysis opens a new door to a *random matrix-improved* analysis framework of computationally efficient methods for machine learning and neural network models.

1.1 Contributions

Our main results are summarized as follows.

1. By considering a high-dimensional Gaussian mixture model for the data, we show (Theorem 1) that for $\mathbf{K}$ defined in (2), $\|\mathbf{K} - \tilde{\mathbf{K}}\| \to 0$ as $n, p \to \infty$, where $\tilde{\mathbf{K}}$ is a random matrix *independent* of the law of $\mathbf{w}$, and depends on the nonlinear $\sigma(\cdot)$ *only* via its generalized Gaussian moments $\mathbb{E}[\sigma'(z)]$ and $\mathbb{E}[\sigma''(z)]$ for $z \sim \mathcal{N}(0, 1)$.

[1] Left- and right-multiplying the kernel matrices by $\mathbf{P}$ is equivalent to center the data in the kernel feature space, which is a common practice in kernel learning and plays a crucial role in multidimensional scaling (Joseph & Myron, 1978) and kernel PCA (Schlkopf et al., 1998). In the remainder of this paper, whenever we use kernel matrices, they are considered to have been centered in this way.

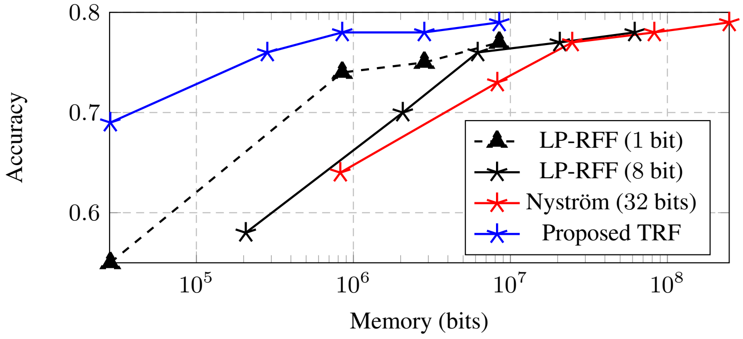

Figure 1: Test accuracy of logistic regression on quantized random features for different number of features $m \in \{10^2, 10^3, 5.10^3, 10^4, 5.10^4\}$, using LP-RFF (8-bit in **black** and 1-bit in dashed **black**) (Zhang et al., 2019), Nyström approximation (32 bits in **red**) (Williams & Seeger, 2001), versus the proposed TRF approach (in **blue**), on the two-class Cov-Type dataset from UCI ML repo, with $n = 418\,000$ training samples, $n_{test} = 116\,000$ test samples, and data dimension $p = 54$.

2. We exploit this result to propose a computationally efficient random features approach, called Ternary Random Features (TRFs), with (asymptotically) the same limiting kernel as *any* random features-type kernel matrices $\mathbf{K}$ of the form (2), while inducing no multiplication and b times less memory storage for its computation, with b the number of bits to store full precision values, e.g., $b = 32$ in the case of a single-precision floating-point format.
3. We provide empirical evidence on various random-features based algorithms, showing the computational and storage advantages of the proposed TRF method, while achieving competitive performances compared to state-of-the-art random features techniques. As a first telling example, in Figure 1 on the Cov-Type dataset from the UCI ML repository,[2] TRFs achieve similar logistic regression performance as the LP-RFF method proposed by Zhang et al. (2019), with 8 times less memory, and 32 times less memory than the Nyström approximation of the Gaussian kernel (using full precision 32 bits) (Williams & Seeger, 2001); see the shift in the x-axis (memory).

1.2 Related work

Random features kernel and random neural networks. Random features methods were first proposed to relieve the computational and storage burden of kernel methods in large-scale problems when the number of training samples n is large (Schölkopf & Smola, 2018; Rahimi & Recht, 2008; Liu et al., 2020). For instance, Random Fourier Features are used to approximate the popular Gaussian kernel, when the number of random features m is sufficiently large (Rahimi & Recht, 2008). Since (deep) modern neural networks are routinely trained with random initialization, random features method is also considered a stylish model to analyze neural networks (Neal, 1996; Williams, 1997; Novak et al., 2018; Matthews et al., 2018; Lee et al., 2017; Garriga-Alonso et al., 2018; Louart et al., 2018). In particular, by focusing on the regime of large n, m, the analysis of random features models led to the so-called *double descent* theory (Advani et al., 2020; Mei & Montanari, 2019; Liao et al., 2020) for neural nets.

Computationally efficient random features methods. In an effort to further reduce the computation and storage costs of random features models, various quantization and binarization methods have been proposed (Goemans & Williamson, 1995; Charikar, 2002; Li & Slawski, 2017; Li & Li, 2019; 2021; Agrawal et al., 2019; Zhang et al., 2019; Liao et al., 2021; Couillet et al., 2021). More precisely, Agrawal et al. (2019) combine RFFs with a data-dependent feature selection approach to reduce the computational cost, while preserving the statistical guarantees of (using the original set of) RFFs. Zhang et al. (2019) propose a low-precision approximation of RFFs to significantly reduce the storage while generalizing as well as full-precision RFFs. Li & Li (2021) design quantization schemes of RFFs for arbitrary choice of the Gaussian kernel parameter. Our work generalizes these

[2] http://archive.ics.uci.edu/ml/index.php

previous efforts by (i) considering a broader family of random features kernels beyond RFFs and (ii) proposing the TRF approach that is both sparse and quantized, while asymptotically yielding the same limiting kernel spectral structure (and thus algorithmic performances (Cortes et al.)).

Random matrix theory and neural networks. Random matrix theory (RMT), as a powerful and flexible tool to investigate the (asymptotic) behavior of large-scale systems, is recently gaining popularity in the analysis of (deep) neural networks. In this respect, Pennington & Worah (2017) derived the eigenvalue distribution of the Conjugate Kernel (CK) in a single-hidden-layer random neural network model. This result was then generalized to a broader class of data distributions (Louart et al., 2018) and to a multi-layer scenario (Benigni & Péché, 2019; Pastur, 2020). Fan & Wang (2020) went beyond the general i.i.d. assumption (on the entries of data vectors) and studied the spectral properties of the CK and neural tangent kernel on data that are approximately "pairwise orthogonal." Our work improves (Fan & Wang, 2020) by studying the random features kernel for more structured GMM data, and is thus more adapted to machine learning applications such as classification. As far as the study of random features kernels under GMM data is concerned, the closest work to ours is (Liao & Couillet, 2018b) where the kernel matrix $\mathbf{K}$ defined in (2) is studied for GMM data, but only for a few specific activation functions and Gaussian $\mathbf{w}$ (see Footnote 4 for a detailed account of the technical differences between this work and (Liao & Couillet, 2018b)). Here, we provide a universal result with respect to the much broader class of activation functions and random $\mathbf{w}$, and propose a computation and storage efficient random features technique well tuned to match the performances of *any* commonly used random features kernels.

1.3 Notations and Organization of the article

In this article, we denote scalars by lowercase letters, vectors by bold lowercase, and matrices by bold uppercase. We denote the transpose operator by $(\cdot)^\mathsf{T}$, we use $\|\cdot\|$ to denote the Euclidean norm for vectors and spectral/operator norm for matrices. For a random variable z, $\mathbb{E}[z]$ denotes the expectation of z. The notation $\delta_{x\in A}$ is the Kronecker delta taking value 1 when $x \in A$ and 0 otherwise. Finally, $\mathbf{1}_p$ and $\mathbf{I}_p$ are respectively the vector of all one's of dimension p and the identity matrix of dimension $p \times p$.

The remainder of the article is structured as follows. In Section 2, we describe the random features model under study along with our working assumptions. We then present our main technical result in Section 3 on the spectral characterization of random features kernel matrices $\mathbf{K}$ and its practical consequences, in particular the design of cost-efficient ternary random features leading asymptotically to the same kernel as any generic random features. We provide empirical evidence in Section 4 showing the computational and storage advantage along with competitive performances compared to state-of-the-art random features approaches. Conclusion and perspective are placed in Section 5.

2 System settings

Let $\mathbf{W} \in \mathbb{R}^{m\times p}$ be a random matrix having i.i.d. entries with zero mean and unit variance. The random features matrix $\mathbf{\Sigma} \in \mathbb{R}^{m\times n}$ of some data $\mathbf{X} \in \mathbb{R}^{p\times n}$ is defined as $\mathbf{\Sigma} \triangleq \sigma(\mathbf{WX})$ for some nonlinear activation function $\sigma : \mathbb{R} \to \mathbb{R}$ applied entry-wise on $\mathbf{WX}$. We denote the associated random features Gram matrix

$$\mathbf{G} \triangleq \frac{1}{m}\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma} = \frac{1}{m}\sigma(\mathbf{WX})^\mathsf{T}\sigma(\mathbf{WX}) = \frac{1}{m}\sum_{t=1}^{m}\sigma\left(\mathbf{X}^\mathsf{T}\mathbf{w}_t\right)\sigma\left(\mathbf{w}_t^\mathsf{T}\mathbf{X}\right) \tag{6}$$

which is a sample mean of the *expected* kernel defined in (1). With $\mathbf{P} \triangleq \mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\mathsf{T}$, we consider the *expected* and *centered* random features kernel matrix $\mathbf{K}$ defined in (2), which plays a fundamental role in various random features kernel-based learning methods such as kernel ridge regression, logistic regression, support vector machines, principal component analysis, or spectral clustering.

Let $\mathbf{x}_1, \cdots, \mathbf{x}_n \in \mathbb{R}^p$ be n independent data vectors belonging to one of K distributional classes $\mathcal{C}_1, \cdots, \mathcal{C}_K$, and class $\mathcal{C}_a$ has cardinality n_a. We assume that $\mathbf{x}_i$ follows a Gaussian Mixture Model (GMM), i.e., for $\mathbf{x}_i \in \mathcal{C}_a$,

$$\mathbf{x}_i = \boldsymbol{\mu}_a/\sqrt{p} + \mathbf{z}_i \tag{7}$$

with $\mathbf{z}_i \sim \mathcal{N}(\mathbf{0}_p, \mathbf{C}_a/p)$ for some mean $\boldsymbol{\mu}_a \in \mathbb{R}^p$ and covariance $\mathbf{C}_a \in \mathbb{R}^{p \times p}$ associated to class $\mathcal{C}_a$.

In high dimensions, under "non triviality assumptions",[3] these data vectors can be shown to be neither very "close" nor very "far" from each other, irrespective of the class they belong to, see (Couillet et al., 2016). We place ourselves under the same non-trivial conditions, by imposing, as in (Couillet et al., 2018), the following growth rate conditions.

Assumption 1 (High-dimensional asymptotics) *As $n \to \infty$, we have (i) $p/n \to c \in (0, \infty)$ and $n_a/n \to c_a \in (0, 1)$; (ii) $\|\boldsymbol{\mu}_a - \boldsymbol{\mu}^\circ\| = O(1)$ for $\boldsymbol{\mu}^\circ = \sum_{a=1}^K \frac{n_a}{n} \boldsymbol{\mu}_a$; (iii) for $\mathbf{C}^\circ = \sum_{a=1}^K \frac{n_a}{n} \mathbf{C}_a$ and $\mathbf{C}_a^\circ = \mathbf{C}_a - \mathbf{C}^\circ$, $\|\mathbf{C}_a\| = O(1)$, $\operatorname{tr}(\mathbf{C}_a^\circ) = O(\sqrt{p})$ and $\operatorname{tr}(\mathbf{C}_a^\circ \mathbf{C}_b) = O(p)$ for $a, b \in \{1, \cdots, K\}$. We denote $\tau \triangleq \operatorname{tr}(\mathbf{C}^\circ)/p$ that is assumed to converge in $(0, \infty)$.*

Remark 1 (Beyond Gaussian mixture data) *While the theoretical results in this paper are derived for Gaussian mixture data in (7), under the non-trivial setting of Assumption 1, we conjecture that they can be extended to a much broader family of data distributions beyond the Gaussian setting, e.g., to the so-called* concentrated random vector *family (Seddik et al., 2020; Louart & Couillet, 2018), under similar non triviality assumptions. See Section A.1 in the appendix for more discussions.*

To cover a large family of random features, we assume that the random projector matrix $\mathbf{W}$ has i.i.d. entries of zero mean, unit variance and bounded fourth-order moment, with no restriction on their particular distribution.

Assumption 2 (On random projection matrix) *The random matrix $\mathbf{W}$ has i.i.d. entries such that $\mathbb{E}[W_{ij}] = 0$, $\mathbb{E}[W_{ij}^2] = 1$ and $\mathbb{E}[W_{ij}^4]$ is finite.*

We consider the family of activation functions $\sigma(\cdot)$ satisfying the following assumption.

Assumption 3 (On activation function) *The function σ is at least twice differentiable (in the sense of distributions when applied to a random variable having a non-degenerate distribution function, see Remark 2 in Section A.1 of the appendix), with $\max\{\mathbb{E}|\sigma(z)|, \mathbb{E}|\sigma^2(z)|, \mathbb{E}|\sigma'(z)|, \mathbb{E}|\sigma''(z)|\} < \lambda$ for some constant $\lambda < \infty$ and $z \sim \mathcal{N}(0, 1)$.*

3 Main result

Our objective is to characterize the high-dimensional spectral behavior of the centered and expected random features kernel matrix $\mathbf{K}$ defined in (2). It turns out, somewhat surprisingly, that under the non-trivial setting of Assumption 1, one may mentally picture high-dimensional vectors as (i) being asymptotically pairwise *orthogonal* (i.e., $\mathbf{x}_i^\mathsf{T} \mathbf{x}_j \to 0$ for $i \neq j$ as $p \to \infty$) and (ii) having asymptotically *equal* (Euclidean) norms (i.e., $\|\mathbf{x}_i\|^2 \to \tau$), *independently* of the underlying class they belong to. As we shall see, this high-dimensional "concentration" of $\mathbf{x}_i^\mathsf{T} \mathbf{x}_j \to \tau \cdot \delta_{i=j}$ plays a crucial role in the Taylor expansion of the (weakly) differentiable nonlinear function σ, which is applied, in the definition of (1), to two *dependent* but asymptotically Gaussian random variables $\sigma(\mathbf{w}^\mathsf{T} \mathbf{x}_i)$ and $\sigma(\mathbf{w}^\mathsf{T} \mathbf{x}_j)$. Due to this asymptotic normality, it is then possible to perform a Gram-Schmidt procedure to work on the nonlinear transformations of two (asymptotically) *independent* Gaussian random variables. This, up to some careful control on the higher-order (but well "concentrated", as a result of the aforementioned high-dimensional "concentration") terms, leads to the following result on the asymptotic behavior of $\mathbf{K}$, the proof of which is given in Section A.3 of the appendix.

Theorem 1 (Asymptotic equivalent of $\mathbf{K}$) *Under Assumption 1-3, for $\mathbf{K}$ defined in (2), as $n \to \infty$,*

$$\|\mathbf{K} - \tilde{\mathbf{K}}\| \to 0,$$

almost surely with

$$\tilde{\mathbf{K}} = \mathbf{P} \left(d_1 \cdot \left(\mathbf{Z} + \mathbf{M} \frac{\mathbf{J}^\mathsf{T}}{\sqrt{p}} \right)^\mathsf{T} \left(\mathbf{Z} + \mathbf{M} \frac{\mathbf{J}^\mathsf{T}}{\sqrt{p}} \right) + d_2 \cdot \mathbf{V} \mathbf{A} \mathbf{V}^\mathsf{T} + d_0 \cdot \mathbf{I}_n \right) \mathbf{P}, \tag{8}$$

[3] That is, when classification is neither too hard nor too easy.

$$\mathbf{V} = \left[\frac{\mathbf{J}}{\sqrt{p}},\ \boldsymbol{\phi} \right] \in \mathbb{R}^{n \times (K+1)}, \quad \mathbf{A} = \begin{bmatrix} \mathbf{t}\mathbf{t}^\mathsf{T} + 2\mathbf{T} & \mathbf{t} \\ \mathbf{t}^\mathsf{T} & 1 \end{bmatrix} \in \mathbb{R}^{(K+1) \times (K+1)}, \quad \mathbf{P} = \mathbf{I}_n - \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^\mathsf{T}$$

where, for $z \sim \mathcal{N}(0,1)$,

$$d_0 = \mathbb{E}[\sigma^2(\sqrt{\tau} z)] - \mathbb{E}[\sigma(\sqrt{\tau} z)]^2 - \tau \mathbb{E}[\sigma'(\sqrt{\tau} z)]^2$$
$$d_1 = \mathbb{E}[\sigma'(\sqrt{\tau} z)]^2, \quad d_2 = \frac{1}{4} \mathbb{E}[\sigma''(\sqrt{\tau} z)]^2$$

and "first-order" random matrix $\mathbf{Z} = [\mathbf{z}_1, \cdots, \mathbf{z}_n] \in \mathbb{R}^{p \times n}$ *as defined in (7), "second-order" random (fluctuation) vector* $\boldsymbol{\phi} = \left\{ \|\mathbf{z}_i\|^2 - \mathbb{E}[\|\mathbf{z}_i\|^2] \right\}_{i=1}^n \in \mathbb{R}^n$*, data statistics*

$$\mathbf{M} = [\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K] \in \mathbb{R}^{p \times K}, \quad \mathbf{t} = \left\{ \frac{\operatorname{tr}(\mathbf{C}_a^\circ)}{\sqrt{p}} \right\}_{a=1}^K \in \mathbb{R}^K, \quad \mathbf{T} = \left\{ \frac{\operatorname{tr} \mathbf{C}_a \mathbf{C}_b}{p} \right\}_{a,b=1}^K \in \mathbb{R}^{K \times K} \tag{9}$$

as well as the class label vector $\mathbf{J} = [\mathbf{j}_1, \cdots, \mathbf{j}_K] \in \mathbb{R}^{n \times K}$ *with* $[\mathbf{j}_a]_i = \delta_{\mathbf{x}_i \in \mathcal{C}_a}$.

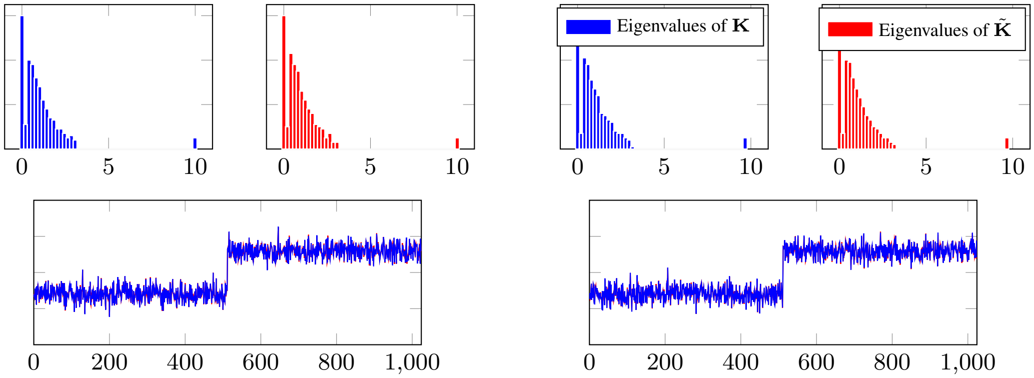

Figure 2: Eigenvalue distribution (**TOP**) and eigenvector associated to the largest eigenvalue (**BOTTOM**) of the expected and centered kernel matrix $\mathbf{K}$ (**blue**) versus its asymptotic equivalent $\tilde{\mathbf{K}}$ (**red**) in Theorem 1, with $\sigma(t) = \max(t, 0)$. (**LEFT**) $\mathbf{W}$ having **Gaussian** entries and (**RIGHT**) $\mathbf{W}$ having **Student-t** entries with 7 degrees of freedom, for two-class GMM data with $\boldsymbol{\mu}_a = [\mathbf{0}_{a-1};\ 4;\ \mathbf{0}_{p-a}], \mathbf{C}_a = (1 + 4(a-1)/\sqrt{p})\mathbf{I}_p$, $p = 512$ and $n = 2048$.

As a direct consequence of Theorem 1, one has, by Weyl's inequality and Davis–Kahan theorem that (i) the difference between each corresponding pair of eigenvalues and (ii) the distance between the "isolated" eigenvectors or the "angle" between the "isolated subspaces" of $\mathbf{K}$ and $\tilde{\mathbf{K}}$ vanish as $n, p \to \infty$. This is numerically confirmed in Figure 2, where one observes a close match between the spectra (eigenvalue "bulk" and isolated eigenvalue-eigenvector pairs) of $\mathbf{K}$ and those of its asymptotic equivalent $\tilde{\mathbf{K}}$ given in Theorem 1, already for n, p only in hundreds. In particular, we compare, in Figure 2, the eigenspectra of $\mathbf{K}$ and $\tilde{\mathbf{K}}$ for Gaussian versus Student-t distributed $\mathbf{W}$, on Gaussian mixture data. A close match of the spectra and the isolated eigen-pairs is observed, irrespective of the distribution of $\mathbf{W}$, as predicted by our theory.

Theorem 1 generalizes (Liao & Couillet, 2018b, Theorem 1) to arbitrary $\mathbf{W}$ and activation function σ satisfying Assumption 2 and 3, respectively.[4] Specifically, the expression of $\tilde{\mathbf{K}}$ (and thus the spectral behavior of $\mathbf{K}$ according to the discussion above) is *universal* with respect to the choice of $\mathbf{W}$ and of the activation function σ, when they are "normalized" to satisfy Assumption 2 and 3. On closer inspection of Theorem 1, we see that the data statistics for classification, i.e., the means ($\mathbf{M}$) and covariances ($\mathbf{t}\mathbf{t}^\mathsf{T}$, $\mathbf{T}$) are respectively weighted by the generalized Gaussian moments of first

[4] From a technical perspective, Theorem 1 crucially differs from (Liao & Couillet, 2018b, Theorem 1) in the fact that the latter relies on the *explicit* forms of the expectation $\mathbf{K}$ in (2), and is thus limited to (i) a few nonlinear σ for which $\mathbf{K}$ can be computed explicitly, see (Liao & Couillet, 2018b, Table 1); and (ii) Gaussian distributed $\mathbf{W}$ in which case the p-dimensional integral can be easily reduced to a two-dimensional one. Here, Theorem 1 holds for a much broader family of random $\mathbf{W}$ and nonlinear σ as long as Assumption 2 and 3 hold.

derivative (d_1) and second derivative (d_2), while the coefficient d_0 merely acts as a regularization term to shift *all* the eigenvalues,[5] and has asymptotically *no* impact on the performance of, e.g., kernel spectral clustering for which only eigenvector structures are exploited.[6]

In the following corollary, we exploit the universal result in Theorem 1 to design the computationally efficient Ternary Random Features (TRFs) with limiting kernel $\mathbf{K}^{ter}$ asymptotically equivalent to *any* random features kernel matrix $\mathbf{K}$ of the form (2), for the high-dimensional GMM under study.

Corollary 1 (Ternary Random Features) *For a given random features kernel matrix $\mathbf{K}$ of the form (2) with $\mathbf{W}$ and nonlinear σ satisfying Assumption 1-3, with associated generalized Gaussian moments d_0, d_1, d_2 defined in Theorem 1, let σ^{ter} be defined in (3) with $s_- = \hat{s}_-$, $s_+ = \hat{s}_+$, and $\hat{s}_-$, $\hat{s}_+$ satisfying the following equations*

$$d_1 = \frac{1}{\pi^2}\left(e^{-\hat{s}_+^2/\tau} + e^{-\hat{s}_-^2/\tau}\right)^2, \quad d_2 = \frac{1}{2\pi\tau^3}\left(\hat{s}_+ e^{-\hat{s}_+^2/\tau} + \hat{s}_- e^{-\hat{s}_-^2/\tau}\right)^2. \tag{10}$$

Define the Ternary Random Features matrix $\mathbf{\Sigma}^{ter} = \sigma^{ter}(\mathbf{W}^{ter}\mathbf{X})$ with $\mathbf{W}^{ter}$ defined in (4) having sparsity level ϵ, the associated Gram matrix $\mathbf{G}^{ter} = \frac{1}{m}(\mathbf{\Sigma}^{ter})^\mathsf{T}\mathbf{\Sigma}^{ter}$ as in (6), and the limiting kernel

$$\mathbf{K}^{ter} \triangleq \mathbf{P}\{\kappa^{ter}(\mathbf{x}_i, \mathbf{x}_j)\}_{i,j=1}^n \mathbf{P} \tag{11}$$

for κ^{ter} defined in (3). Then, as $n \to \infty$ there exists $\lambda \in \mathbb{R}$ such that[7]

$$\|\mathbf{K} - \mathbf{K}^{ter} - \lambda\mathbf{P}\| \to 0,$$

almost surely.

Note that the system of equations in (10) defining $\hat{s}_-$ and $\hat{s}_+$ is a function of the key parameter $\tau = \operatorname{tr}\mathbf{C}^\circ/p$, which can be consistently estimated from the data; see Algorithm 1 below and a proof in Lemma 1 of the appendix. This makes the proposed TRFs and the associated limiting kernel $\mathbf{K}^{ter}$ data-statistics-dependent. Yet, the system of equations (10) does not have a closed-form analytical solution and might have multiple solutions on the real line; we practically solve it using numerical least squares methods, by gradually enlarging the search range (from say $[-1, 1]$) until a solution is found. The details of the proposed TRFs are described in Algorithm 1.

Algorithm 1 Ternary Random Features

```
Input: Data $\mathbf{X}$ and level of sparsity $\epsilon \in [0, 1)$.
Output: Ternary Random Features $\mathbf{\Sigma}^{ter}$ and Gram matrix $\mathbf{G}^{ter}$.
Estimate $\tau$ as $\hat{\tau} = \frac{1}{n}\sum_{i=1}^n \|\mathbf{x}_i\|^2$.
Solve for thresholds $\hat{s}_-$, $\hat{s}_+$ using (10), which defines $\sigma^{ter}$ via (3).
Construct a random matrix $\mathbf{W}^{ter} \in \mathbb{R}^{m \times p}$ having i.i.d. entries distributed according to (4).
Compute $\mathbf{\Sigma}^{ter} = \sigma^{ter}(\mathbf{W}^{ter}\mathbf{X})$ and then TRFs Gram matrix $\mathbf{G}^{ter} = \frac{1}{m}(\mathbf{\Sigma}^{ter})^\mathsf{T}\mathbf{\Sigma}^{ter}$ as in (6).
```

Computational and storage complexity For $\mathbf{W} \in \mathbb{R}^{m \times p}$ a random matrix with i.i.d. $\mathcal{N}(0, 1)$ entries and $\mathbf{W}^{ter} \in \mathbb{R}^{m \times p}$ with i.i.d. entries satisfying (4) with sparsity level $\epsilon \in [0, 1)$, let $\mathbf{G} = \frac{1}{m}\sigma(\mathbf{W}\mathbf{X})^\mathsf{T}\sigma(\mathbf{W}\mathbf{X})$ for some given smooth function σ (e.g., sine and cosine in the case of random Fourier features (Rahimi & Recht, 2008)) and $\mathbf{G}^{ter} = \frac{1}{m}\sigma^{ter}(\mathbf{W}^{ter}\mathbf{X})^\mathsf{T}\sigma^{ter}(\mathbf{W}^{ter}\mathbf{X})$ with data matrix $\mathbf{X} \in \mathbb{R}^{p \times n}$. The computation of $\mathbf{G}$ thus requires $O(mnp)$ multiplications and $O(mnp)$ additions, while the construction of $\mathbf{G}^{ter}$ requires no multiplication and only $O(\epsilon mnp)$ additions. In terms of storage, it requires a factor of $b = 32$ times more bits to store $\sigma(\mathbf{W}\mathbf{X})$ when computing $\mathbf{G}$ compared to storing $\sigma^{ter}(\mathbf{W}^{ter}\mathbf{X})$ when computing $\mathbf{G}^{ter}$ (assuming full precision numbers are stored using $b = 32$ bits).

The computationally and storage efficient TRFs $\sigma^{ter}(\mathbf{W}^{ter}\mathbf{X})$ can then be used instead of the "expensive" random features $\sigma(\mathbf{W}\mathbf{X})$ and leads (asymptotically) to the same performance as the

[5] This can be seen as another manifestation of the *implicit regularization* in high-dimensional kernel and random features ridge regression (Jacot et al., 2020; Derezinski et al., 2020; Liu et al., 2021).

[6] We provide in Table 1 (Section A.4 of the appendix) the calculation of the Gaussian moments d_0, d_1, d_2 for various commonly used activation functions in random features and neural network contexts.

[7] The parameter λ characterizes the *possibly* different d_0 between $\mathbf{K}$ and $\mathbf{K}^{ter}$.

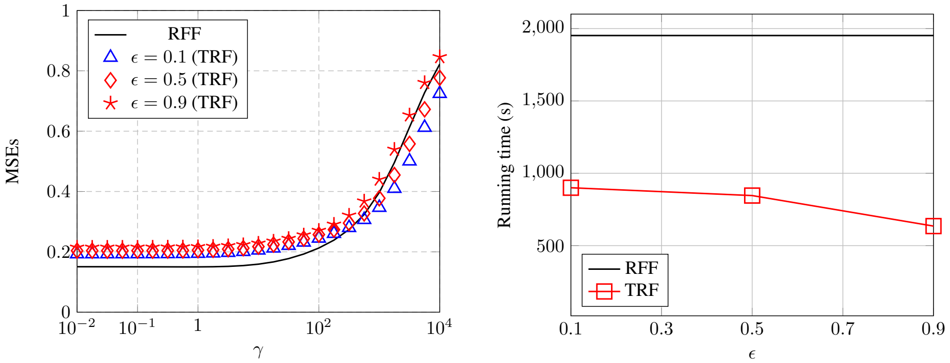

Figure 3: Testing mean squared errors (MSEs, **LEFT**) and running time (**RIGHT**) of kernel ridge regression as a function of regularization parameter $\gamma, p = 512, n = 1024, n_{test} = 512, m = 5.10^4$. Ternary function (with thresholds s_-, s_+ chosen to match the Gaussian moments d_1, d_2 of RFFs) with $\mathbf{W}$ distributed according to (4) with $\epsilon = [0.1, 0.5, 0.9]$, versus RFFs on a 2-class MNIST dataset – digits $(7, 9)$. Results averaged over 5 independent runs.

latter on downstream tasks, at least for GMM data, according to Theorem 1. In the following section, we will provide empirical results showing that (i) this "performance match" between TRFs and any random features of the form (1) is *not* limited to Gaussian data and empirically holds when applied on popular real-world datasets such as MNIST (LeCun et al., 1998), Cov-Type and Census datasets (from the UCI ML repo), as well as DNN-features of CIFAR10 data in Section A.5 of the appendix; and (ii) due to the competitive performance of TRFs with respect to standard random features approaches, when compared to state-of-the-art random feature compression/quantization techniques (for which a "performance-complexity tradoff" generally arises, that is, as one compresses more the original random features, the performance decays), TRFs yield significantly better performances for a given storage/computational budget.

4 Experiments

The experiments in this section and Section A.5 of the appendix are performed on a Ubuntu 18.04 machine with Intel(R) Core(TM) i9-9900X CPU @ 3.50GHz and 64 GB of RAM.

4.1 TRFs match the performance of popular kernels with considerably less budget

We first consider ridge regression with random features Gram matrix $\mathbf{G}$ on MNIST data (LeCun et al., 1998) in Figure 3. We compare (i) RFFs with $\sigma(t) = [\cos(t),\ \sin(t)]$ and Gaussian $W_{ij} \sim \mathcal{N}(0, 1)$ to (ii) the proposed TRF method with $\sigma^{ter}(t)$ in (3) and ternary random projection matrix $\mathbf{W}^{ter}$ defined in (4), with different sparsity levels ϵ. The thresholds s_-, s_+ of σ^{ter} are tuned in such away that the generalized Gaussian moments d_1 and d_2 are matched with those of RFFs[8], as described in Corollary 1 and Algorithm 1. Figure 3 displays the test mean squared errors as a function of the ridge regularization parameter γ for our TRF method with different sparsity levels ϵ compared to the RFF method. Note that despite 90% sparsity in the projection matrix $\mathbf{W}$, virtually no performance loss is incurred compared with RFFs. This is further confirmed in the right hand side of Figure 3 which shows the gains in running time[9] when using TRFs. These experiments show that the proposed TRF approach yields similar performance as popular random features such as RFFs, with a significant gain in terms of computation and storage.

[8] For given $\mathbf{x}_i, \mathbf{x}_j$, we use $[\cos(\mathbf{W}\mathbf{x}_i),\ \sin(\mathbf{W}\mathbf{x}_j)]$ as the random Fourier features so that the (i, j) entry of the corresponding random features Gram matrix is $\cos(\mathbf{W}\mathbf{x}_i)^\mathsf{T} \cos(\mathbf{W}\mathbf{x}_j) + \sin(\mathbf{W}\mathbf{x}_i)^\mathsf{T} \sin(\mathbf{W}\mathbf{x}_j)$ (Rahimi & Recht, 2008). The generalized Gaussian moments of the RFFs are thus the sum of the d_1's corresponding to sin and cos functions, and the sum of the corresponding d_2's.

[9] The running time is taken as the total clock-time for the whole ridge regression solver including the random features calculation.

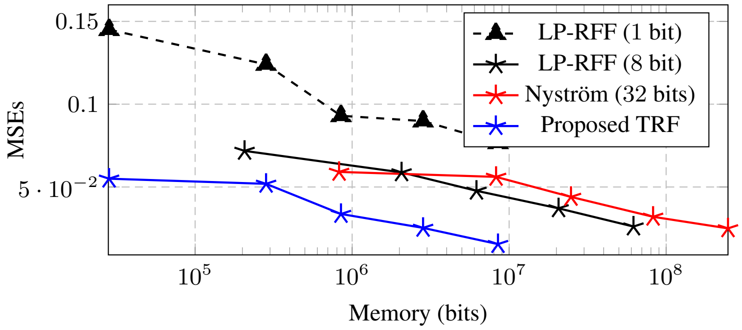

Figure 4: Test mean square errors of ridge regression on quantized random features for different numbers of features $m \in \{5.10^2, 10^3, 5.10^3, 10^4, 5.10^4\}$, using LP-RFF (Zhang et al., 2019), Nyström approximation (Williams & Seeger, 2001), versus the proposed TRF approach, on the Census dataset from UCI ML repo, with $n = 16\,000$ training samples, $n_{test} = 2\,000$ test samples, and data dimension $p = 119$.

4.2 Computational and storage gains – Comparisons to state-of-the-art

In this section, we compare TRFs with state-of-the-art quantized or non-quantized random features methods, for both random features-based logistic and ridge regressions. Specifically, in Figure 1, we compare logistic regression performance of TRFs versus the Low Precision Random Fourier Features (LP-RFFs) proposed by Zhang et al. (2019), on the Cov-Type dataset from UCI ML repo. As in Section 4.1, the TRFs are tuned to match the limiting Gaussian kernel of RFFs. For a single datum $\mathbf{x} \in \mathbb{R}^p$, the associated TRFs $\sigma^{ter}(\mathbf{W}^{ter}\mathbf{x}) \in \mathbb{R}^m$ use m bits for storage while LP-RFFs use $32m$ bits. We follow the same protocol as in (Zhang et al., 2019) and use SGD with mini-batch size 250 in training logistic regressor and $\{1, 8\}$ bits precision for the LP-RFF approach. Figure 1 compares the logistic regression test accuracy as a function of the total memory budget for LP-RFF (1 bit and 8 bits) and the Nyström approximation of Gaussian kernel matrices (using full precision 32 bits) (Williams & Seeger, 2001) (see also Table 1 in (Zhang et al., 2019)), versus the proposed TRFs approach. As seen from the shift in the x-axis (memory), by using $8\times$ less memory than LP-RFFs and $32\times$ less memory than the Nyström method, TRFs achieve a superior generalization performance on the logistic regression task. The same comparisons are performed in Figure 4 on a random features ridge regression task for the Census dataset (Rahimi & Recht, 2008). Figure 4 shows that TRFs outperform alternative approaches in terms of test mean square errors, with a significant gain in memory.

5 Conclusion

Our large dimensional spectral analysis of the random features kernel matrices $\mathbf{K}$ reveals that its spectral properties only depend on the non-linear activation through the corresponding generalized Gaussian moments and are universal with respect to zero-mean and unit-variance random projection vectors. This allowed us to design the new random features technique TRF which turns both the random weights $\mathbf{W}$ and the activations $\sigma(\mathbf{WX})$ into ternary integers, thereby allowing to only perform addition operations and only storing 1 bit for the activations. This drastically saves the storage and computation of random features while preserving the performances on downtream tasks with respect to their counterpart expensive kernels. A possible future direction is to also ternarize the input data $\mathbf{X}$ so as to work only with integers. We expect that the extension of Theorem 1 to data following a mixture of concentrated random vectors (Seddik et al., 2020) to be the path leading to that total ternarization of random features models. We leave this for future work. Our article comes along with the work in (Couillet et al., 2021) as first steps in re-designing machine learning algorithms using Random Matrix Theory, in order to be able to perform computations on massive data on desktop computers instead of relying on high consuming giant servers. Although our work sets the algorithmic way of making large gains in practice, a huge effort needs to be made in re-designing software libraries for numerical operations to smartly exploit the sparse and (finite set) integer structure in large matrices. This will facilitate the embedding of (still) efficient algorithms into small devices.

REFERENCES

Madhu S. Advani, Andrew M. Saxe, and Haim Sompolinsky. High-dimensional dynamics of generalization error in neural networks. *Neural Networks*, 132:428–446, 2020. ISSN 0893-6080. doi: 10.1016/j.neunet.2020.08.022.

Raj Agrawal, Trevor Campbell, Jonathan Huggins, and Tamara Broderick. Data-dependent compression of random features for large-scale kernel approximation. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1822–1831. PMLR, 2019.

Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. *arXiv preprint arXiv:1811.04918*, 2018.

Sanjeev Arora, Simon Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning*, pp. 322–332. PMLR, 2019.

Zhidong Bai and Jack W. Silverstein. *Spectral Analysis of Large Dimensional Random Matrices*, volume 20 of *Springer Series in Statistics*. Springer-Verlag New York, 2 edition, 2010. ISBN 9781441906601. doi: 10.1007/978-1-4419-0661-8.

Lucas Benigni and Sandrine Péché. Eigenvalue distribution of nonlinear models of random matrices. *arXiv preprint arXiv:1904.03090*, 2019.

Alberto Bietti and Julien Mairal. On the inductive bias of neural tangent kernels. *arXiv preprint arXiv:1905.12173*, 2019.

Patrick Billingsley. *Probability and Measure*. Wiley Series in Probability and Statistics. John Wiley & Sons, Ltd, 3 edition, 2012. ISBN 9781118122372. URL `https://www.wiley.com/en-us/Probability+and+Measure%2C+Anniversary+Edition-p-9781118122372`.

Moses S Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pp. 380–388, 2002.

Lénaïc Chizat, Edouard Oyallon, and Francis Bach. On Lazy Training in Differentiable Programming. In *Advances in Neural Information Processing Systems*, volume 32 of *NIPS'19*, pp. 2937–2947. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper/2019/file/ae614c557843b1df326cb29c57225459-Paper.pdf`.

Youngmin Cho. *Kernel methods for deep learning*. University of California, San Diego, 2012.

Corinna Cortes, Mehryar Mohri, and Ameet Talwalkar. On the Impact of Kernel Approximation on Learning Accuracy. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 113–120, Chia Laguna Resort, Sardinia, Italy. PMLR. URL `https://proceedings.mlr.press/v9/cortes10a.html`.

Couillet, Cinar Romain, Gaussier Y., and Imran M. E. Word representations concentrate and this is good news! *In Proceedings of the 24th Conference on Computational Natural Language Learning*, pp. 325–334, 2020, November.

Romain Couillet, Florent Benaych-Georges, et al. Kernel spectral clustering of large dimensional data. *Electronic Journal of Statistics*, 10(1):1393–1454, 2016.

Romain Couillet, Zhenyu Liao, and Xiaoyi Mai. Classification asymptotics in the random matrix regime. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pp. 1875–1879. IEEE, 2018.

Romain Couillet, Florent Chatelain, and Nicolas Le Bihan. Two-way kernel matrix puncturing: towards resource-efficient pca and spectral clustering. *arXiv preprint arXiv:2102.12293*, 2021.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Michal Derezinski, Feynman T Liang, and Michael W Mahoney. Exact expressions for double descent and implicit regularization via surrogate random design. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5152–5164. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/37740d59bb0eb7b4493725b2e0e5289b-Paper.pdf.

Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In *International Conference on Machine Learning*, pp. 1675–1685. PMLR, 2019.

Zhou Fan and Zhichao Wang. Spectra of the conjugate kernel and neural tangent kernel for linear-width neural networks. *arXiv preprint arXiv:2005.11879*, 2020.

Friedrich Gerard Friedlander, G Friedlander, Mark Suresh Joshi, M Joshi, and Mohan C Joshi. *Introduction to the Theory of Distributions*. Cambridge University Press, 1998.

Adrià Garriga-Alonso, Carl Edward Rasmussen, and Laurence Aitchison. Deep convolutional networks as shallow gaussian processes. *arXiv preprint arXiv:1808.05587*, 2018.

Michel X Goemans and David P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6): 1115–1145, 1995.

Yunchao Gong, Liu Liu, Ming Yang, and Lubomir Bourdev. Compressing deep convolutional networks using vector quantization. *arXiv preprint arXiv:1412.6115*, 2014.

Suyog Gupta, Ankur Agrawal, Kailash Gopalakrishnan, and Pritish Narayanan. Deep learning with limited numerical precision. In *International conference on machine learning*, pp. 1737–1746. PMLR, 2015.

Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015a.

Song Han, Jeff Pool, John Tran, and William Dally. Learning both Weights and Connections for Efficient Neural Network. In *Advances in Neural Information Processing Systems*, volume 28 of *NIPS'15*. Curran Associates, Inc., 2015b. URL https://proceedings.neurips.cc/paper/2015/file/ae0eb3eed39d2bcef4622b2499a05fe6-Paper.pdf.

Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized Neural Networks. In *Advances in Neural Information Processing Systems*, volume 29 of *NIPS'16*, pp. 4107–4115. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/file/d8330f857a17c53d217014ee776bfd50-Paper.pdf.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. PMLR, 2015.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *arXiv preprint arXiv:1806.07572*, 2018.

Arthur Jacot, Berfin Simsek, Francesco Spadaro, Clement Hongler, and Franck Gabriel. Implicit regularization of random feature models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4631–4640. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/jacot20a.html.

Kruskal Joseph and Wish Myron. *Multidimensional Scaling*. 1978. ISBN 9780803909403. doi: 10.4135/9781412985130. URL https://methods.sagepub.com/book/multidimensional-scaling.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Pei Ling Lai and Colin Fyfe. Kernel and nonlinear canonical correlation analysis. *International Journal of Neural Systems*, 10(05):365–377, 2000.

Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. ISSN 0018-9219. doi: 10.1109/5.726791.

Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. *arXiv preprint arXiv:1711.00165*, 2017.

Jaehoon Lee, Lechao Xiao, Samuel S Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *arXiv preprint arXiv:1902.06720*, 2019.

Ping Li and Martin Slawski. Simple strategies for recovering inner products from coarsely quantized random projections. *Advances in Neural Information Processing Systems*, 30:4567–4576, 2017.

Xiaoyun Li and Ping Li. Random projections with asymmetric quantization. *Advances in Neural Information Processing Systems*, 32:10858–10867, 2019.

Xiaoyun Li and Ping Li. Quantization algorithms for random fourier features. *arXiv preprint arXiv:2102.13079*, 2021.

Zhenyu Liao and Romain Couillet. The dynamics of learning: A random matrix approach. In *International Conference on Machine Learning*, pp. 3072–3081. PMLR, 2018a.

Zhenyu Liao and Romain Couillet. On the spectrum of random features maps of high dimensional data. In *International Conference on Machine Learning*, pp. 3063–3071. PMLR, 2018b.

Zhenyu Liao, Romain Couillet, and Michael W Mahoney. A random matrix analysis of random fourier features: beyond the gaussian kernel, a precise phase transition, and the corresponding double descent. 33:13939–13950, 2020. URL `https://proceedings.neurips.cc/paper/2020/file/a03fa30821986dff10fc66647c84c9c3-Paper.pdf`.

Zhenyu Liao, Romain Couillet, and Michael W. Mahoney. Sparse quantized spectral clustering. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=pBqLS-7KYAF`.

Zhouhan Lin, Matthieu Courbariaux, Roland Memisevic, and Yoshua Bengio. Neural networks with few multiplications. *arXiv preprint arXiv:1510.03009*, 2015.

Baoyuan Liu, Min Wang, Hassan Foroosh, Marshall Tappen, and Marianna Pensky. Sparse convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 806–814, 2015.

Fanghui Liu, Xiaolin Huang, Yudong Chen, and Johan A. K. Suykens. Random Features for Kernel Approximation: A Survey on Algorithms, Theory, and Beyond. *arXiv*, 2020. URL `https://arxiv.org/abs/2004.11154`.

Fanghui Liu, Zhenyu Liao, and Johan Suykens. Kernel regression in high dimensions: Refined analysis beyond double descent. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 649–657. PMLR, 13–15 Apr 2021. URL `https://proceedings.mlr.press/v130/liu21b.html`.

Cosme Louart and Romain Couillet. Concentration of measure and large random matrices with an application to sample covariance matrices. *arXiv preprint arXiv:1805.08295*, 2018.

Cosme Louart, Zhenyu Liao, Romain Couillet, et al. A random matrix approach to neural networks. *The Annals of Applied Probability*, 28(2):1190–1248, 2018.

Yueming Lyu. Spherical structured feature maps for kernel approximation. In *International Conference on Machine Learning*, pp. 2256–2264. PMLR, 2017.

Alexander G de G Matthews, Mark Rowland, Jiri Hron, Richard E Turner, and Zoubin Ghahramani. Gaussian process behaviour in wide deep neural networks. *arXiv preprint arXiv:1804.11271*, 2018.

Song Mei and Andrea Montanari. The generalization error of random features regression: Precise asymptotics and double descent curve. *arXiv preprint arXiv:1908.05355*, 2019.

Radford M Neal. Priors for infinite networks. In *Bayesian Learning for Neural Networks*, pp. 29–53. Springer, 1996.

Roman Novak, Lechao Xiao, Jaehoon Lee, Yasaman Bahri, Greg Yang, Jiri Hron, Daniel A Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Bayesian deep convolutional networks with many channels are gaussian processes. *arXiv preprint arXiv:1810.05148*, 2018.

Leonid Pastur. On random matrices arising in deep neural networks. gaussian case. *arXiv preprint arXiv:2001.06188*, 2020.

Jeffrey Pennington and Pratik Worah. Nonlinear random matrix theory for deep learning. 2017.

Danil Prokhorov. Ijcnn 2001 neural network competition. 2001.

Haotong Qin, Ruihao Gong, Xianglong Liu, Xiao Bai, Jingkuan Song, and Nicu Sebe. Binary neural networks: A survey. *Pattern Recognition*, 105:107281, 2020.

Ali Rahimi and Benjamin Recht. Random Features for Large-Scale Kernel Machines. In *Advances in Neural Information Processing Systems*, volume 20 of *NIPS'08*, pp. 1177–1184. Curran Associates, Inc., 2008. URL `https://proceedings.neurips.cc/paper/2007/file/013a006f03dbc5392effeb8f18fda755-Paper.pdf`.

Bernhard Schlkopf, Alexander Smola, and Klaus-Robert Mller. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, 10(5):1299–1319, 1998. ISSN 0899-7667. doi: 10.1162/089976698300017467.

Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *International conference on artificial neural networks*, pp. 583–588. Springer, 1997.

Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press, 2018. ISBN 9780262256933. doi: 10.7551/mitpress/4175.001.0001.

Mohamed El Amine Seddik, Mohamed Tamaazousti, and Romain Couillet. A kernel random matrix-based approach for sparse pca. In *International Conference on Learning Representations (ICLR)*, 2019.

Mohamed El Amine Seddik, Cosme Louart, Mohamed Tamaazousti, and Romain Couillet. Random matrix theory proves that deep learning representations of gan-data behave as gaussian mixtures. In *International Conference on Machine Learning*, pp. 8573–8582. PMLR, 2020.

Elias M Stein and Rami Shakarchi. *Functional Analysis, Introduction to Further Topics in Analysis*. 2012. doi: 10.1515/9781400840557-005.

Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

Vladimir Vovk. Kernel ridge regression. In *Empirical inference*, pp. 105–116. Springer, 2013.

Christopher Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. In *Proceedings of the 14th annual conference on neural information processing systems*, number CONF, pp. 682–688, 2001.

Christopher KI Williams. Computing with infinite networks. *Advances in neural information processing systems*, pp. 295–301, 1997.

Felix Xinnan X Yu, Ananda Theertha Suresh, Krzysztof M Choromanski, Daniel N Holtmann-Rice, and Sanjiv Kumar. Orthogonal random features. *Advances in neural information processing systems*, 29:1975–1983, 2016.

Jian Zhang, Avner May, Tri Dao, and Christopher Ré. Low-precision random fourier features for memory-constrained kernel approximation. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1264–1274. PMLR, 2019.

Chenzhuo Zhu, Song Han, Huizi Mao, and William J Dally. Trained ternary quantization. *arXiv preprint arXiv:1612.01064*, 2016.

A Appendix

A.1 Notes on the working Assumptions

For completeness, let us redefine the working settings of the paper.

Let $\mathbf{x}_1, \cdots, \mathbf{x}_n \in \mathbb{R}^p$ be n independent vectors belonging to one of K distributional classes $\mathcal{C}_1, \cdots, \mathcal{C}_K$. Class a has cardinality n_a, and we assume that $\mathbf{x}_i$ follows a Gaussian Mixture Model (GMM), i.e., for $\mathbf{x}_i \in \mathcal{C}_a$,

$$\mathbf{x}_i = \boldsymbol{\mu}_a/\sqrt{p} + \mathbf{z}_i \tag{12}$$

with $\mathbf{z}_i \sim \mathcal{N}(\mathbf{0}_p, \mathbf{C}_a/p)$ for some mean $\boldsymbol{\mu}_a \in \mathbb{R}^p$ and covariance $\mathbf{C}_a \in \mathbb{R}^{p\times p}$ associated to class $\mathcal{C}_a$.

As motivated in the core of the article, we are working in the following non-trivial regime in the high dimensional classification as described by the following growth rate conditions:

Assumption 4 (High-dimensional asymptotics) *As $n \to \infty$, we have (i) $p/n \to c \in (0, \infty)$ and $n_a/n \to c_a \in (0, 1)$; (ii) $\|\boldsymbol{\mu}_a\| = O(1)$; (iii) for $\mathbf{C}^\circ = \sum_{a=1}^K \frac{n_a}{n}\mathbf{C}_a$ and $\mathbf{C}_a^\circ = \mathbf{C}_a - \mathbf{C}^\circ$, then $\|\mathbf{C}_a\| = O(1)$, $\operatorname{tr}(\mathbf{C}_a^\circ) = O(\sqrt{p})$ and $\operatorname{tr}(\mathbf{C}_a\mathbf{C}_b) = O(p)$ for $a, b \in \{1, \cdots, K\}$. We denote $\tau \triangleq \operatorname{tr}(\mathbf{C}^\circ)/p$ that is assumed to converge in $(0, \infty)$.*

Beyond Gaussian mixtures While the theoretical results in this paper are derived for Gaussian mixture data in (12) under the non-trivial setting in Assumption 4, we conjecture that they can be extended to a much broader family of data distributions beyond the Gaussian setting, e.g., to the so-called *concentrated random vector* family (Seddik et al., 2020), under similar non triviality assumptions. It has been shown in (Seddik et al., 2020) that images generated by a Generative Adversarial Networks, which look similar to real-world images, are Lipshitz transformations of Gaussian vectors and can thus be modelled as Concentrated random variables. The same was shown experimentally for CNN representations of real images (Seddik et al., 2019; 2020) as well as words embeddings in Natural Language Processing (Couillet et al., 2020, November). Additionally, (Seddik et al., 2019; 2020) show that Gram matrices of independent concentrated data from a mixture models asymptotically behave as if the data were drawn from a Gaussian Mixture Model.

As for the random projector matrix, we assume that the matrix $\mathbf{W}$ has i.i.d. entries of zero mean, unit variance and bounded fourth-order moment, with no restriction on their particular distribution as follows.

Assumption 5 (On random projection matrix) *The random matrix $\mathbf{W}$ has i.i.d. entries such that $\mathbb{E}[W_{ij}] = 0$, $\mathbb{E}[W_{ij}^2] = 1$ and $\mathbb{E}[W_{ij}^4]$ is finite.*

We consider the family of activation functions $\sigma(\cdot)$ satifying the following assumption.

Assumption 6 (On activation function) *The function σ is at least twice differentiable (in the sense of distributions when applied to a random variable having a non-degenerate distribution function), with $\max\{\mathbb{E}|\sigma(z)|, \mathbb{E}|\sigma^2(z)|, \mathbb{E}|\sigma'(z)|, \mathbb{E}|\sigma''(z)|\} < \lambda$ for some constant $\lambda < \infty$ and $z \sim \mathcal{N}(0, 1)$.*

Remark 2 (Activation functions not differentiable everywhere) *Some popular activation functions used in machine learning such as ReLu, Sign, Absolute value, etc., are not differentiable everywhere. For those functions, we will use a derivative in the sense of distributions (Friedlander et al., 1998). A distribution g is a continuous linear functional on the set $\mathcal{D}$ of infinitely differentiable* *functions with bounded support*

$$\begin{aligned} g &: \mathcal{D} \to \mathbb{R} \\ \phi &\mapsto g(\phi) = \int_{-\infty}^{+\infty} g(x)\phi(x)\,\mathrm{d}x. \end{aligned}$$

The distributional derivative $g'(\phi)$ is defined such that $g'(\phi) = -g(\phi')$. In particular, we will be interested here in the expectation of the derivatives of the activation function with respect to the Gaussian measure i.e., $\int_{-\infty}^{+\infty} \sigma'(x) e^{-x^2/2}\,\mathrm{d}x$. Following the previous definition, we have in this particular case

$$\int_{-\infty}^{+\infty} \sigma'(x) e^{-x^2/2}\,\mathrm{d}x = \int_{-\infty}^{+\infty} x\sigma(x) e^{-x^2/2}\,\mathrm{d}x$$

which can be evaluated by some integration by parts. This is also refereed to the as "weak derivative" in the functional analysis literature (Stein & Shakarchi, 2012).

A.2 Auxilliary results and proofs

Lemma 1 (Consistent estimation of τ) *Let Assumption 4 hold and define $\tau \triangleq \mathrm{tr}\mathbf{C}^\circ/p$. Then as $n \to \infty$, with probability 1,*

$$\frac{1}{n}\sum_{i=1}^n \|\mathbf{x}_i\|^2 - \tau \to 0$$

Proof 1 (Proof of Lemma 1) *From equation 7, we have that*

$$\frac{1}{n}\sum_{i=1}^n \|\mathbf{x}_i\|^2 = \frac{1}{n}\sum_{a=1}^K\sum_{i=1}^n \frac{1}{p}\|\boldsymbol{\mu}_a\|^2 - \frac{2}{\sqrt{p}}\boldsymbol{\mu}_a^\mathsf{T}\mathbf{z}_i + \frac{1}{n}\sum_{i=1}^n \|\mathbf{z}_i\|^2. \tag{13}$$

From Assumption 4, we have that $\frac{1}{n}\sum_{a=1}^K\sum_{i=1}^n \frac{1}{p}\|\boldsymbol{\mu}_a\|^2 = O(p^{-1})$. The second term of equation 13 $\frac{2}{\sqrt{p}}\boldsymbol{\mu}_a^\mathsf{T}\mathbf{z}_i$ is a weighted sum of independent zero mean random variables; it thus vanishes with probability 1 as $n, p \to \infty$ by a mere application of Chebyshev's inequality and the Borell Cantelli lemma. Finally, using the strong law of large numbers on the last term of equation 13, we have $\frac{1}{n}\sum_{i=1}^n \|\mathbf{z}_i\|^2 - \tau \to 0$ almost surely. This concludes the proof.

A.3 Proof of Theorem 1

Let us define $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n] \in \mathbb{R}^{p\times n}$ and define

$$\mathbf{\Sigma} = \sigma(\mathbf{W}\mathbf{X})$$

where $\mathbf{W} = [\mathbf{w}_1, \cdots, \mathbf{w}_m]^\mathsf{T} \in \mathbb{R}^{m\times p}$ with $\mathbf{W}$ satisfying Assumption 5, and σ a function satisfying Assumption 6. We consider the gram matrix

$$\mathbf{G} = \frac{1}{m}\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma}$$

whose entries are given by

$$G_{ij} = \frac{1}{m}\sum_{k=1}^m \sigma(\mathbf{w}_k^\mathsf{T}\mathbf{x}_i)\sigma(\mathbf{w}_k^\mathsf{T}\mathbf{x}_j).$$

We are interested in computing

$$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbb{E}_{\mathbf{w}\sim\mathbf{w}_1}\sigma(\mathbf{w}^\mathsf{T}\mathbf{x}_i)\sigma(\mathbf{w}^\mathsf{T}\mathbf{x}_j)$$

under Assumptions- 4- 5- 6.

Under Assumption 4, we have

$$\mathbf{x}_i^\mathsf{T}\mathbf{x}_j = \underbrace{\mathbf{z}_i^\mathsf{T}\mathbf{z}_j}_{O(p^{-\frac{1}{2}})} + \underbrace{\boldsymbol{\mu}_a^\mathsf{T}\boldsymbol{\mu}_b/p + \boldsymbol{\mu}_a^\mathsf{T}\mathbf{w}_j/\sqrt{p} + \boldsymbol{\mu}_b^\mathsf{T}\mathbf{w}_i/\sqrt{p}}_{O(p^{-1})} \tag{14}$$

and

$$\|\mathbf{x}_i\|^2 = \underbrace{\tau}_{O(1)} + \underbrace{tr(\mathbf{C}_a^\circ)/p + \Phi_i}_{O(p^{-\frac{1}{2}})} + \underbrace{\|\boldsymbol{\mu}_a^2\|/p + 2\boldsymbol{\mu}_a^\mathsf{T}\mathbf{z}_i/\sqrt{p}}_{O(p^{-1})} \tag{15}$$

where $\Phi_i \triangleq (\|\mathbf{z}_i\|^2 - \mathbb{E}[\|\mathbf{z}_i\|^2])$.

It can be checked that, for random vector $\mathbf{w} \in \mathbb{R}^p$ having i.i.d. entries with $\mathbb{E}[w_i] = 0$ and $\mathbb{E}[w_i^2] = 1$, we have, conditioned on $\mathbf{x}_i$, that $\mathbb{E}_\mathbf{w}[(\mathbf{w}^\mathsf{T}\mathbf{x}_i)^2] = \|\mathbf{x}_i\|^2$ and

$$\mathbb{E}_\mathbf{w}[(\mathbf{w}^\mathsf{T}\mathbf{x}_i)^4] = (m_4 - 3)\|\mathbf{x}_i\|^2 + 2\|\mathbf{x}_i\|^4. \tag{16}$$

where $m_4 = \mathbb{E}[\|\mathbf{w}\|^4]$. From the trace lemma, (Bai & Silverstein, 2010, Lemma B.26), one has that

$$\mathbf{x}_i^\mathsf{T}\mathbf{x}_i - \frac{1}{p}\mathrm{tr}\mathbf{C}_a \to 0 \tag{17}$$

almost surely as $p \to \infty$, for $\mathbf{x}_i \in \mathcal{C}_a$. Thus, under Assumption 4 we have in particular $\mathbf{x}_i^\mathsf{T}\mathbf{x}_i - \mathrm{tr}\mathbf{C}^\circ/p \to 0$ holds almost surely, regardless of the class of $\mathbf{x}_i$. It then follows from Lyapunov CLT (see, e.g., (Billingsley, 2012, Theorem 27.3)) that, under Assumption 4 and 5, $(\mathbf{w}^\mathsf{T}\mathbf{x}_i, \mathbf{w}^\mathsf{T}\mathbf{x}_j)$ is asymptotically bivariate Gaussian. We can thus perform a Gram-Schmidt orthogonalization procedure for some standard Gaussian variables $\xi_a, \xi_b \sim \mathcal{N}(0, 1)$. We have

$$\begin{aligned}\mathbf{w}^\mathsf{T}\mathbf{x}_i &= u_a\xi_a \\ \mathbf{w}^\mathsf{T}\mathbf{x}_j &= v_b\xi_a + u_b\xi_b\end{aligned}$$

with

$$\begin{aligned}u_a &= \|x_i\| \\ v_b &= \frac{\mathbf{x}_i^\mathsf{T}\mathbf{x}_j}{\|x_i\|} \\ u_b &= \sqrt{\|x_j\|^2 - \frac{(\mathbf{x}_i^\mathsf{T}\mathbf{x}_j)^2}{\|x_i\|^2}}.\end{aligned}$$

Let us denote $\zeta_a = \mathbf{w}^\mathsf{T}\mathbf{x}_i$ and $\zeta_b = \mathbf{w}^\mathsf{T}\mathbf{x}_j$. Since we have that $\|\mathbf{x}_i\| = \sqrt{\tau} + o(1)$ and $\|\mathbf{x}_j\| = \sqrt{\tau} + o(1)$ (from Equation equation 15), we can perform a Taylor expansion of $\sigma(\zeta_a)$ (resb. $\sigma(\zeta_b)$) around $\sqrt{\tau}\xi_a$ (resp. $\sqrt{\tau}\xi_b$) giving

$$\begin{aligned}\sigma(\zeta_a) &= \sigma(\sqrt{\tau}\xi_a) + \sigma'(\sqrt{\tau}\xi_a)(\zeta_a - \sqrt{\tau}\xi_a) + \frac{1}{2}\sigma''(\xi_a)(\zeta_a - \sqrt{\tau}\xi_a)^2 + o((\zeta_a - \sqrt{\tau}\xi_a)^2) \\ \sigma(\zeta_b) &= \sigma(\sqrt{\tau}\xi_b) + \sigma'(\sqrt{\tau}\xi_b)(\zeta_b - \sqrt{\tau}\xi_b) + \frac{1}{2}\sigma''(\sqrt{\tau}\xi_b)(\zeta_b - \sqrt{\tau}\xi_b)^2 + o((\zeta_b - \sqrt{\tau}\xi_b)^2).\end{aligned}$$

We then have

$$\begin{aligned}K(\mathbf{x}_i, \mathbf{x}_j) &= \mathbb{E}[\sigma(\zeta_a)\sigma(\zeta_b)] \\ &= \mathbb{E}[\sigma(\sqrt{\tau}\xi_a)\sigma(\sqrt{\tau}\xi_b)] + \mathbb{E}[\sigma(\sqrt{\tau}\xi_a)\sigma'(\sqrt{\tau}\xi_b)(\zeta_b - \sqrt{\tau}\xi_b)] + \frac{1}{2}\mathbb{E}[\sigma(\sqrt{\tau}\xi_a)\sigma''(\sqrt{\tau}\xi_b)(\zeta_b - \sqrt{\tau}\xi_b)^2] \\ &+ \mathbb{E}[\sigma'(\sqrt{\tau}\xi_a)(\zeta_a - \sqrt{\tau}\xi_a)\sigma(\sqrt{\tau}\xi_b)] + \mathbb{E}[\sigma'(\sqrt{\tau}\xi_a)(\zeta_a - \sqrt{\tau}\xi_a)\sigma'(\sqrt{\tau}\xi_b)(\zeta_b - \sqrt{\tau}\xi_b)] \\ &+ \frac{1}{2}\mathbb{E}[\sigma'(\xi_a)(\zeta_a - \sqrt{\tau}\xi_a)\sigma''(\xi_b)(\zeta_b - \sqrt{\tau}\xi_b)^2] + \frac{1}{2}\mathbb{E}[\sigma''(\xi_a)(\zeta_a - \sqrt{\tau}\xi_a)^2\sigma(\sqrt{\tau}\xi_b)] \\ &+ \frac{1}{2}\mathbb{E}[\sigma''(\sqrt{\tau}\xi_a)(\zeta_a - \sqrt{\tau}\xi_a)^2\sigma'(\sqrt{\tau}\xi_b)(\zeta_b - \sqrt{\tau}\xi_b)] \\ &+ \frac{1}{4}\mathbb{E}[\sigma''(\sqrt{\tau}\xi_a)(\zeta_a - \sqrt{\tau}\xi_a)^2\sigma''(\sqrt{\tau}\xi_b)(\zeta_b - \sqrt{\tau}\xi_b)^2].\end{aligned}$$

Using the independence between ξ_a, ξ_b along with the following

- $\zeta_a - \sqrt{\tau}\xi_a = (\frac{u_a}{\sqrt{\tau}} - 1)\sqrt{\tau}\xi_a$
- $\zeta_b - \sqrt{\tau}\xi_b = (\frac{u_b}{\sqrt{\tau}} - 1)\sqrt{\tau}\xi_b + \frac{v_b}{\sqrt{\tau}}\sqrt{\tau}\xi_a$

we get for $\xi \in N(0,1)$,

$$\begin{aligned}
K(\mathbf{x}_i,\mathbf{x}_j) = \Bigg[& \left(\mathbb{E}[\sigma(\sqrt{\tau}\xi)]\right)^2 + \left(\mathbb{E}[\sigma(\sqrt{\tau}\xi)]\mathbb{E}[\sqrt{\tau}\xi\sigma'(\sqrt{\tau}\xi)]\right)\left(\frac{u_b}{\sqrt{\tau}}-1\right) + \left(\mathbb{E}[\sigma'(\sqrt{\tau}\xi)]\mathbb{E}[\sqrt{\tau}\xi\sigma(\sqrt{\tau}\xi)]\right)\frac{v_b}{\sqrt{\tau}} \\
& + \left(\mathbb{E}[\sqrt{\tau}\xi\sigma'(\sqrt{\tau}\xi)]^2\right)\left((\frac{u_a}{\sqrt{\tau}}-1)(\frac{u_b}{\sqrt{\tau}}-1)\right) + \frac{\left(\mathbb{E}[\sigma(\sqrt{\tau}\xi)]\mathbb{E}[\tau\xi^2\sigma''(\sqrt{\tau}\xi)]\right)}{2}\left((u_b-1)^2\right) \\
& + \frac{\left(\mathbb{E}[\sqrt{\tau}\xi\sigma'(\sqrt{\tau}\xi)]\mathbb{E}[\tau\xi^2\sigma''(\sqrt{\tau}\xi)]\right)}{2}\left(\frac{u_a}{\sqrt{\tau}}-1\right)\left(\frac{u_b}{\sqrt{\tau}}-1\right)^2 \\
& + \frac{\left(\mathbb{E}[\sqrt{\tau}\xi\sigma'(\sqrt{\tau}\xi)]\mathbb{E}[\tau\xi^2\sigma''(\sqrt{\tau}\xi)]\right)}{2}\left(\frac{u_a}{\sqrt{\tau}}-1\right)^2\left(\frac{u_b}{\sqrt{\tau}}-1\right) \\
& + \frac{\left(\mathbb{E}[\tau\xi^2\sigma''(\sqrt{\tau}\xi)]^2\right)}{4}\left(\frac{u_a}{\sqrt{\tau}}-1\right)^2\left(\frac{u_b}{\sqrt{\tau}}-1\right)^2 + \left(\mathbb{E}[\sqrt{\tau}\xi\sigma(\sqrt{\tau}\xi)]\mathbb{E}[\sqrt{\tau}\xi\sigma''(\sqrt{\tau}\xi)]\right)\frac{v_b}{\sqrt{\tau}}\left(\frac{u_b}{\sqrt{\tau}}-1\right) \\
& + \frac{\left(\mathbb{E}[\sigma''(\sqrt{\tau}\xi)]\mathbb{E}[\tau\xi^2\sigma''(\sqrt{\tau}\xi)]\right)}{2}\left(\frac{v_b}{\sqrt{\tau}}\right)^2 + \left(\mathbb{E}[\sigma(\sqrt{\tau}\xi)]\mathbb{E}[\sqrt{\tau}\xi\sigma'(\sqrt{\tau}\xi)]\right)\left(\frac{u_a}{\sqrt{\tau}}-1\right) \\
& + \left(\mathbb{E}[\sigma'(\sqrt{\tau}\xi)]\mathbb{E}[\xi^2\sigma'(\sqrt{\tau}\xi)]\right)\frac{v_b}{\sqrt{\tau}}\left(\frac{u_a}{\sqrt{\tau}}-1\right) + \frac{\left(\mathbb{E}[\sigma''(\sqrt{\tau}\xi)]\mathbb{E}[\tau\sqrt{\tau}\xi^3\sigma'(\sqrt{\tau}\xi)]\right)}{2}\left(\frac{v_b}{\sqrt{\tau}}\right)^2\left(\frac{u_a}{\sqrt{\tau}}-1\right) \\
& + \frac{\left(\mathbb{E}[\sigma'(\sqrt{\tau}\xi)]\mathbb{E}[\tau\xi^2\sigma'(\sqrt{\tau}\xi)]\right)}{2}\frac{v_b}{\sqrt{\tau}}\left(\frac{u_a}{\sqrt{\tau}}-1\right)\left(\frac{u_b}{\sqrt{\tau}}-1\right) + \frac{\left(\mathbb{E}[\sigma(\sqrt{\tau}\xi)]\mathbb{E}[\tau\xi^2\sigma''(\sqrt{\tau}\xi)]\right)}{2}\left(\frac{u_a}{\sqrt{\tau}}-1\right)^2 \\
& + \frac{\left(\mathbb{E}[\sigma'(\sqrt{\tau}\xi)]\mathbb{E}[\tau\sqrt{\tau}\xi^3\sigma''(\sqrt{\tau}\xi)]\right)}{2}\frac{v_b}{\sqrt{\tau}}\left(\frac{u_a}{\sqrt{\tau}}-1\right)^2 \\
& + \frac{\left(\mathbb{E}[\sigma''(\sqrt{\tau}\xi)]\mathbb{E}[\tau^2\xi^4\sigma''(\sqrt{\tau}\xi)]\right)}{4}\left(\frac{v_b}{\sqrt{\tau}}\right)^2\left(\frac{u_a}{\sqrt{\tau}}-1\right)^2 \\
& + \frac{\left(\mathbb{E}[\sqrt{\tau}\xi\sigma''(\sqrt{\tau}\xi)]\mathbb{E}[\tau\sqrt{\tau}\xi^3\sigma''(\sqrt{\tau}\xi)]\right)}{2}\frac{v_b}{\sqrt{\tau}}\left(\frac{u_b}{\sqrt{\tau}}-1\right)\left(\frac{u_a}{\sqrt{\tau}}-1\right)^2\Bigg]. \qquad (18)
\end{aligned}$$

Since $|\mathbf{x}_i^\mathsf{T}\mathbf{x}_j| \leq \epsilon$ for a sufficiently small ϵ, (following from Equation equation 14) we have

$$\begin{aligned}
\frac{u_b}{\sqrt{\tau}}-1 &= \frac{1}{\sqrt{\tau}}\sqrt{\|x_j\|^2-\frac{(\mathbf{x}_i^\mathsf{T}\mathbf{x}_j)^2}{\|x_i\|^2}}-1 = \left(\frac{\|x_j\|}{\sqrt{\tau}}-1\right)-\frac{1}{2\sqrt{\tau}\|x_j\|}\frac{(\mathbf{x}_i^\mathsf{T}\mathbf{x}_j)^2}{\|x_i\|^2} \\
\left(\frac{u_b}{\sqrt{\tau}}-1\right)^2 &= \left(\frac{1}{\sqrt{\tau}}\sqrt{\|x_j\|^2-\frac{(\mathbf{x}_i^\mathsf{T}\mathbf{x}_j)^2}{\|x_i\|^2\|x_j\|^2}}-1\right)^2 = \left(\frac{\|x_j\|}{\sqrt{\tau}}-1\right)^2-\frac{\frac{\|x_j\|}{\sqrt{\tau}}-1}{\|x_j\|}\frac{(\mathbf{x}_i^\mathsf{T}\mathbf{x}_j)^2}{\|x_i\|^2} \\
\frac{u_a}{\sqrt{\tau}}-1 &= \frac{\|x_i\|}{\sqrt{\tau}}-1.
\end{aligned}$$

We thus have

$$\begin{aligned}
\frac{v_b}{\sqrt{\tau}} &= \frac{\mathbf{x}_i^\mathsf{T}\mathbf{x}_j}{\sqrt{\tau}\|\mathbf{x}_i\|} = \frac{1}{\tau}\left(\underbrace{\mathbf{z}_i^\mathsf{T}\mathbf{z}_j}_{O(p^{-\frac{1}{2}})}+\underbrace{\left(\frac{\boldsymbol{\mu}_a^\mathsf{T}\boldsymbol{\mu}_b}{p}+\frac{\boldsymbol{\mu}_a^\mathsf{T}\mathbf{w}_j}{\sqrt{p}}+\frac{\boldsymbol{\mu}_b^\mathsf{T}\mathbf{w}_i}{\sqrt{p}}\right)}_{O(p^{-1})}\right)+O(p^{-\frac{3}{2}}) \\
\left(\frac{u_a}{\sqrt{\tau}}-1\right) &= \frac{\|\mathbf{x}_i\|}{\sqrt{\tau}}-1 = \underbrace{\frac{1}{2\tau}\left(\frac{tr(\mathbf{C}_a^\circ)}{p}+\Phi_i\right)}_{O(p^{-\frac{1}{2}})}+\underbrace{\frac{1}{2\tau}\left(\frac{\|\boldsymbol{\mu}_a\|^2}{p}+2\boldsymbol{\mu}_a^\mathsf{T}\frac{\mathbf{z}_i}{\sqrt{p}}\right)}_{O(p^{-1})}+O(p^{-\frac{3}{2}})
\end{aligned}$$

$$\left(\frac{u_a}{\sqrt{\tau}}-1\right)\left(\frac{u_b}{\sqrt{\tau}}-1\right)=\left(\frac{\|\mathbf{x}_i\|}{\sqrt{\tau}}-1\right)\left(\frac{\|\mathbf{x}_j\|}{\sqrt{\tau}}-1\right)=\underbrace{\frac{\left(\frac{tr(\mathbf{C}_a^\circ)}{p}+\Phi_i\right)\left(\frac{tr(\mathbf{C}_b^\circ)}{p}+\Phi_j\right)}{4\tau^2}}_{O(p^{-\frac{1}{2}})}+O(p^{-\frac{3}{2}})$$

$$\left(\frac{v_b}{\sqrt{\tau}}\right)^2=\frac{1}{\tau^2}(\mathbf{z}_i^\intercal\mathbf{z}_j)^2+O(p^{-\frac{3}{2}}). \tag{19}$$

All other terms in equation 18 vanish asymptotically. We thus get

$$\begin{aligned}K(\mathbf{x}_i,\mathbf{x}_j)=\Bigg[&\left(\mathbb{E}[\sigma(\sqrt{\tau}\xi)]\right)^2+\left(\mathbb{E}[\sigma(\sqrt{\tau}\xi)]\mathbb{E}[\sqrt{\tau}\xi\sigma'(\sqrt{\tau}\xi)]\right)\left(\frac{u_b}{\sqrt{\tau}}-1\right)+\left(\mathbb{E}[\sigma'(\sqrt{\tau}\xi)]\mathbb{E}[\sqrt{\tau}\xi\sigma(\sqrt{\tau}\xi)]\right)\frac{v_b}{\sqrt{\tau}}+\\&+\left(\mathbb{E}[\sqrt{\tau}\xi\sigma'(\sqrt{\tau}\xi)]^2\right)\left((\frac{u_a}{\sqrt{\tau}}-1)(\frac{u_b}{\sqrt{\tau}}-1)\right)+\frac{\left(\mathbb{E}[\sigma''(\sqrt{\tau}\xi)]\mathbb{E}[\tau\xi^2\sigma(\sqrt{\tau}\xi)]\right)}{2}\left(\frac{v_b}{\sqrt{\tau}}\right)^2\\&+\left(\mathbb{E}[\sigma(\sqrt{\tau}\xi)]\mathbb{E}[\sqrt{\tau}\xi\sigma'(\sqrt{\tau}\xi)]\right)\left(\frac{u_a}{\sqrt{\tau}}-1\right)\Bigg].\end{aligned} \tag{20}$$

Plugging the terms in equation 19 into equation 20 and rearranging we get the result in Theorem 1.

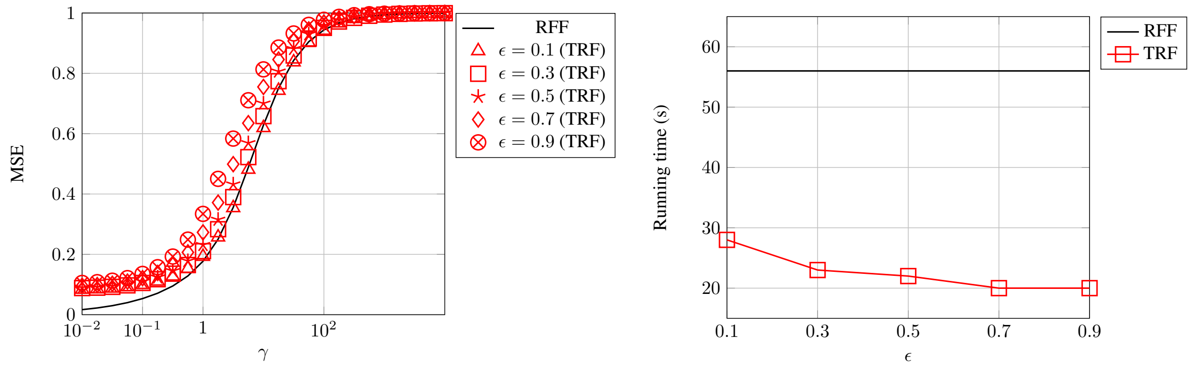

Figure 5: Testing MSE of kernel ridge regression as a function of regularization parameter γ, $p = 512, n = 1024, n_{test} = 512, m = 512$. Ternary function (with thresholds s_-, s_+ chosen to match gaussian moments of $[\cos, \sin]$ function) with $\mathbf{W}$ distributed according to equation 4 with $\epsilon = [0.1, 0.3, 0.5, 0.7]$ (in **red**). versus RFF (in solid **black**)- GMM dataset with $\boldsymbol{\mu}_a = [\mathbf{0}_{a-1}; 4; \mathbf{0}_{p-a}], \mathbf{C}_a = (1 + 4(a-1)/\sqrt{p})\mathbf{I}_p, p = 512, n = 2048.)$

A.4 Gaussian moments of popular activation functions

We provide in Table 1 the calculation of the Generalized Gaussian moments d_0, d_1, d_2 for popular activation functions used in machine learning.

Table 1: Values of d_0, d_1, d_2, d_3 for different activation functions.

$\sigma(t)$	d_0	d_1	d_2
$\lvert t \rvert$	$\tau\left(1 - \frac{2}{\pi}\right)$	0	$\frac{1}{2\pi\tau}$
$\max(0, \xi)$	$\frac{\tau}{2}\left(\frac{1}{2} - \frac{1}{\pi}\right)$	$\frac{1}{4}$	$\frac{1}{8\pi\tau}$
$a_+ \max(0, t) + a_- \max(0, -t)$	$\tau\left(a_+ + a_-\right)^2 \left(\frac{\pi-2}{4\pi}\right)$	$\frac{(a_+ - a_-)^2}{4}$	$\frac{(a_+ + a_-)^2}{8\pi\tau}$
$a_2 t^2 + a_1 t + a_0$	$2\tau^2 a_2^2$	a_1^2	a_2^2
$\exp(t)$	$\frac{1}{\sqrt{2\tau+1}} - \frac{1}{\tau+1}$	0	$\frac{1}{4(\tau+1)^3}$
$\cos(t)$	$\frac{1+e^{-2\tau}}{2} - e^{-\tau}$	0	$\frac{e^{-\tau}}{4}$
$\sin(t)$	$\frac{1-e^{-2\tau}}{2} - \tau e^{-\tau}$	$e^{-\tau}$	0
t	0	1	0
$sign(t)$	$1 - \frac{2}{\pi}$	$\frac{2}{\pi\tau}$	0
$1_{t>0}$	$\frac{1}{4} - \frac{1}{2\pi}$	$\frac{1}{2\pi\tau}$	0

A.5 Additional experiments

Random features based Ridge regression To complement the experiments in Section 4.1, we provide in Figures 5- 6- 7- 8- 9, the test mean square error with increasing number of random features $m \in \{512, 4096, 10^4\}$ for GMM data and $m \in \{512, 10^4\}$ for MNIST data.

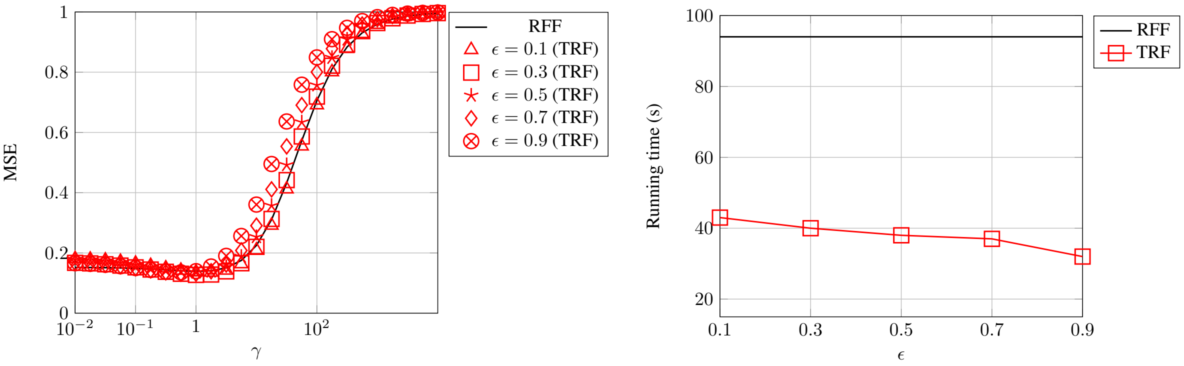

Figure 6: Testing MSE of kernel ridge regression as a function of regularization parameter γ, $p = 512, n = 1024, n_{test} = 512, m = 4096$. Ternary function (with thresholds s_-, s_+ chosen to match gaussian moments of $[\cos, \sin]$ function) with $\mathbf{W}$ distributed according to equation 4 with $\epsilon = [0.1, 0.3, 0.5, 0.7]$ (in **red**). versus RFF (in solid **black**)- GMM dataset with $\boldsymbol{\mu}_a = [\mathbf{0}_{a-1}; 4; \mathbf{0}_{p-a}], \mathbf{C}_a = (1 + 4(a-1)/\sqrt{p})\mathbf{I}_p, p = 512, n = 2048.$)

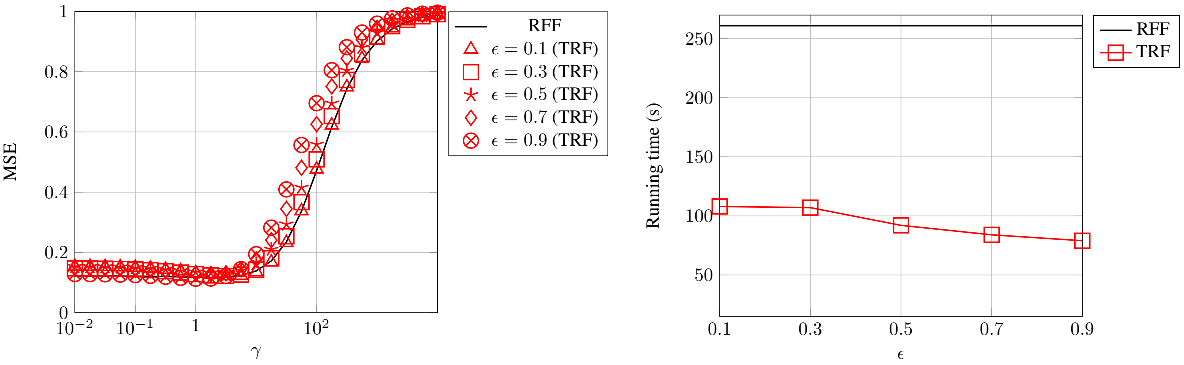

Figure 7: Testing MSE of kernel ridge regression as a function of regularization parameter γ, $p = 512, n = 1024, n_{test} = 512, m = 10^4$. Ternary function (with thresholds s_-, s_+ chosen to match gaussian moments of $[\cos, \sin]$ function) with $\mathbf{W}$ distributed according to equation 4 with $\epsilon = [0.1, 0.3, 0.5, 0.7]$ (in **red**). versus RFF (in solid **black**)- GMM dataset with $\boldsymbol{\mu}_a = [\mathbf{0}_{a-1}; 4; \mathbf{0}_{p-a}], \mathbf{C}_a = (1 + 4(a-1)/\sqrt{p})\mathbf{I}_p, p = 512, n = 2048.$)

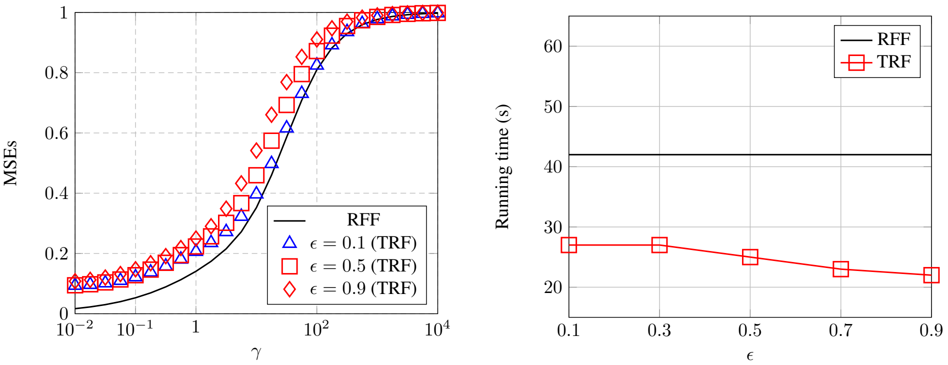

Figure 8: Testing mean squared errors (MSEs, **LEFT**) and running time (**RIGHT**) of kernel ridge regression as a function of regularization parameter γ, $p = 512, n = 1024, n_{test} = 512, m = 512$. Ternary function (with thresholds s_-, s_+ chosen to match the Gaussian moments d_1, d_2 of $[\cos, \ \sin]$ function) with $\mathbf{W}$ distributed according to (4) with $\epsilon = [0.1, 0.3, 0.5, , 0.7, 0.9]$, versus RFFs (in solid **black**) on MNIST dataset 2 classes - digits $(7, 9)$- results averaged over 5 independent runs.

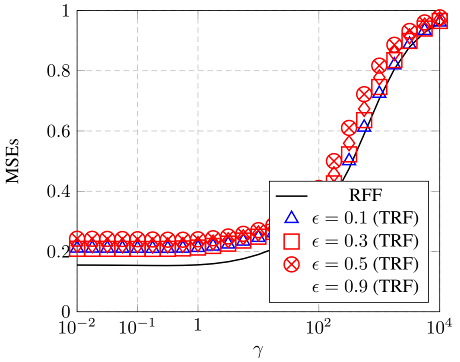

Figure 9: Testing mean squared errors (MSEs, **LEFT**) and running time (**RIGHT**) of kernel ridge regression as a function of regularization parameter γ, $p = 512, n = 1024, n_{test} = 512, m = 10^4$. Ternary function (with thresholds s_-, s_+ chosen to match the Gaussian moments d_1, d_2 of $[\cos, \ \sin]$ function) with $\mathbf{W}$ distributed according to (4) with $\epsilon = [0.1, 0.3, 0.5, , 0.7, 0.9]$, versus RFFs (in solid **black**) on MNIST dataset 2 classes - digits $(7, 9)$- results averaged over 5 independent runs.

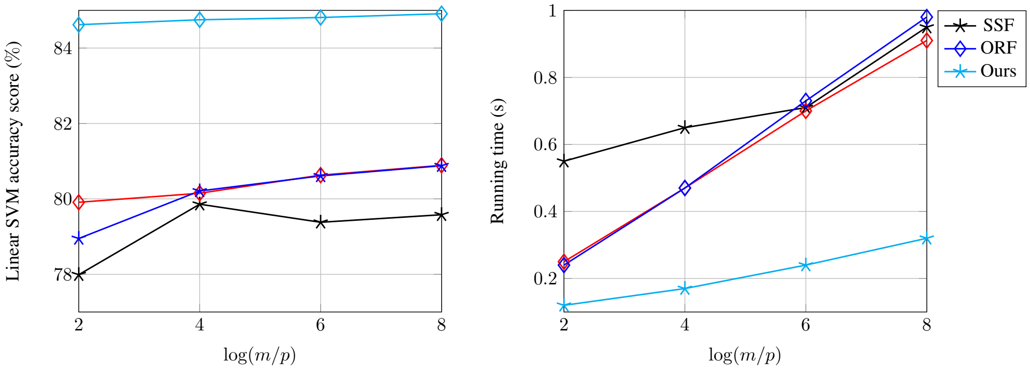

Figure 10: Test accuracy using libsvm on different state-of-the-art random features kernels. a8a (UCI) dataset. Number of training samples $n = 22696$ - number of test samples $n_t = 9865$, varying ratio $\log m/p$ number of random features over dimension $p = 123$

Random features based Support Vector Machine We empirically evaluate the classification performance of various random features approximation algorithms, on several benchmark datasets. We compared the different algorithms (ORF (Yu et al., 2016), SSF (Lyu, 2017)) on 3 datasets (IJCNN1 (Prokhorov, 2001), cov-type, a8a from the UCI ML repository) considered in (Liu et al., 2020), with our TRF method where we choose the thresholds coefficients s_-, s_+ according to Algorithm 1 (to match the Generalized Gaussian moments of $[\cos, \sin]$). Figure 10 shows the results for the a8a dataset, Figure 11 for the IJCNN1 dataset and Figure 12 for the Covtype dataset. The lower running time along with higher SVM test accuracy indicates the superiority of our theory-inspired TRF method over the other approximation kernels.

Comparison OF TRF with equivalent 0^{th} order Arc-cosine kernel (with ReLU function) We consider Support Vector Machine (SVM) classification with random features Gram matrix $\mathbf{G}$ on Fashion MNIST data (LeCun et al., 1998) and on VGG16 embeddings of CIFAR10 (Krizhevsky et al., 2009) and Imagenet (Deng et al., 2009) datasets in Figures 14 -13 -15 respectively. We use VGG16 with batch normalization (Ioffe & Szegedy, 2015) pre-trained on ImageNet (Deng et al., 2009) as a feature extractor. We fine-tune this model on the CIFAR10 dataset with 240 epochs and a mini-batch size 64 with a SGD optimizer with momentum 0.9 and an initial learning rate of 0.1. We then extract the output of the first fully connected layer of the classifier as our features. For Imagenet, we directly extract the features from the pretrained model. We compare (i) RF with $\sigma(t) = \max(t, 0)$ (ReLU) and Gaussian $W_{ij} \sim \mathcal{N}(0, 1)$ (known as the 0th order Arc-cosine kernel) to (ii) the proposed TRF

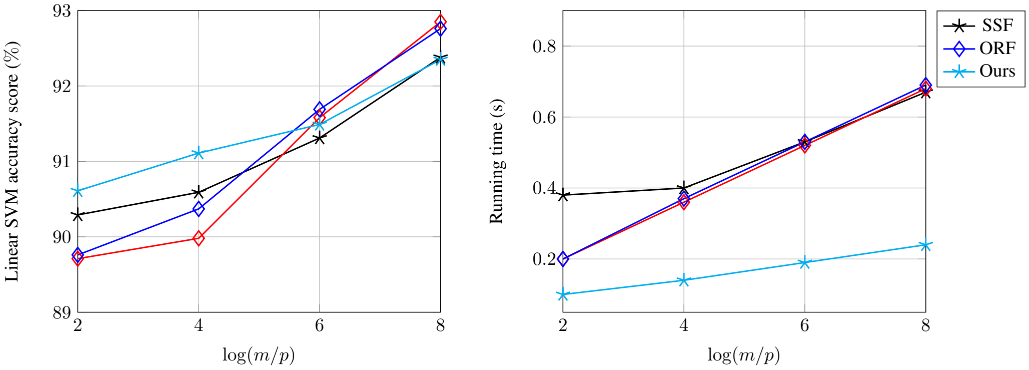

Figure 11: Test accuracy using libsvm on different state-of-the-art random features kernels. IJCNN1 (Prokhorov, 2001) dataset. Number of training samples $n = 49990$ - number of test samples $n_t = 91701$, varying ratio $\log m/p$ number of random features over dimension $p = 22$

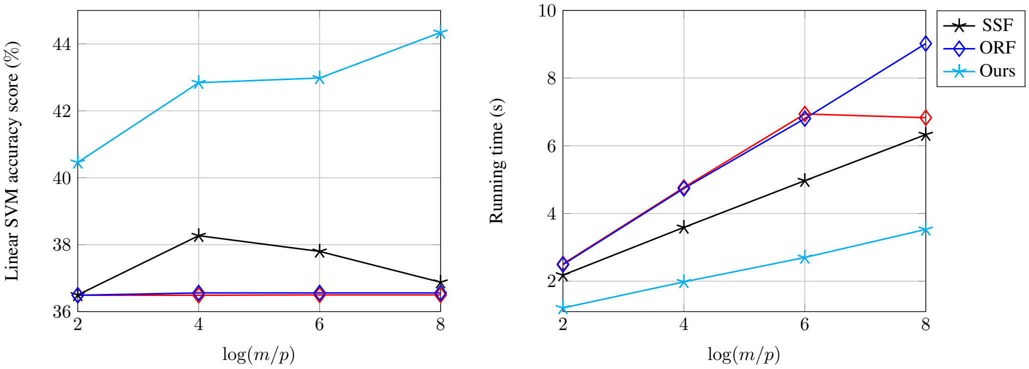

Figure 12: Test accuracy using libsvm on different state-of-the-art random features kernels. Number of training samples $n = 49990$ - number of test samples $n_t = 91701$, varying ratio $\log m/p$ number of random features over dimension $p = 22$

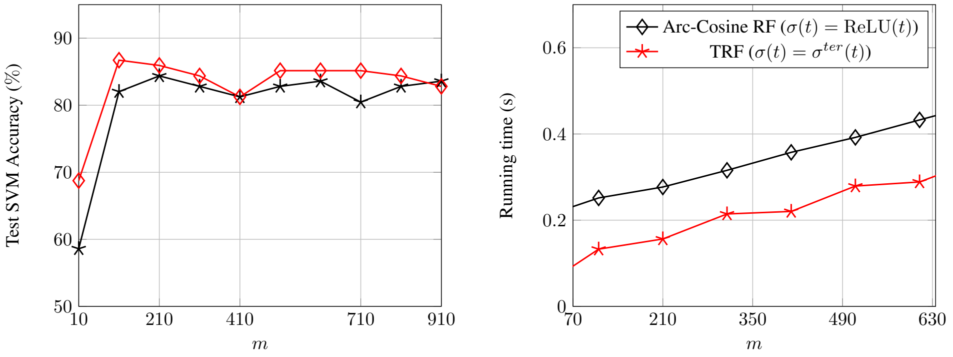

Figure 13: SVM test accuracy using different kernels. VGG-16 Embeddings of CIFAR10 dataset (Number of features $p = 4096$). Number of samples $n = 1024$ fixed, varying number of random features from $m = 10$ to $m = 1800$.

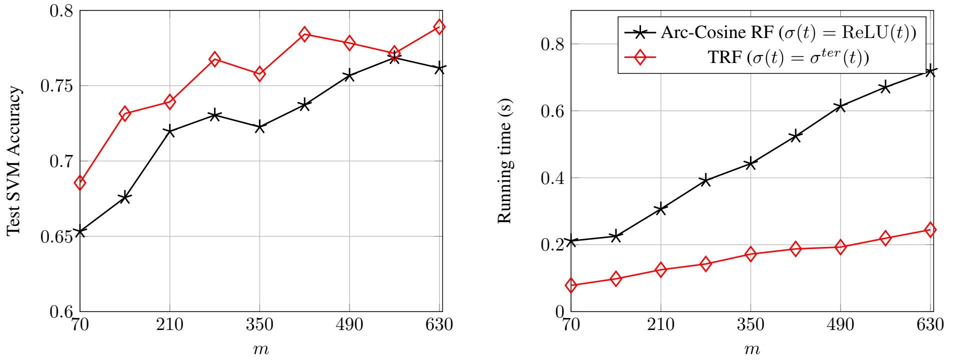

Figure 14: SVM test accuracy using different kernels. Raw Fashion-MNIST dataset (Number of features $p = 784$). Number of samples $n = 1024$ fixed, varying number of random features from $m = 70$ to $m = 700$. Baseline (Linear SVM) compared to random projection methods, SVM on (ReLU and Sparse)

method with $\sigma^{ter}(t)$ in (3) and ternary random projection matrix $\mathbf{W}^{ter}$ defined in (4). The thresholds s_-, s_+ of σ^{ter} are tuned in such away that the generalized Gaussian moments d_1 and d_2 are matched with those of ReLU (see Table 1), as described in Corollary 1 and Algorithm 1. Figures 14 -13 -15 display the test SVM accuracy as a function of the number of random features, for our TRF method compared to the random features corresponding to the Arc-Cosine kernel. We observe similar test performances with the two kernels while having a computation and storage gains for the ternary kernel.

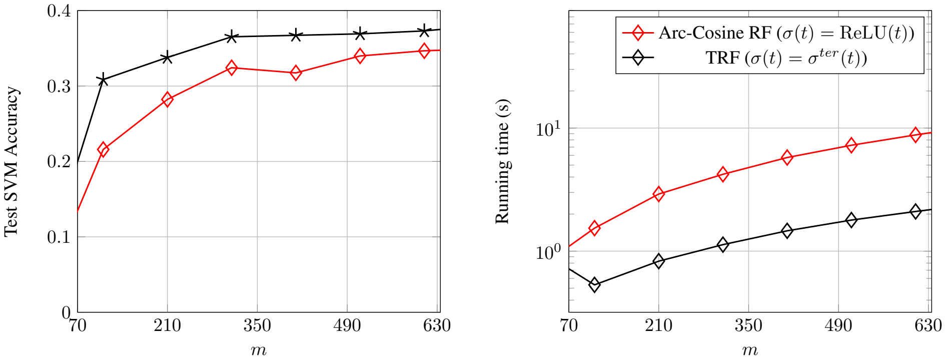

Figure 15: SVM test accuracy using different kernels. LeNet-64 Embeddings of Imagenet dataset (Number of features $p = 2018$). Number of samples $n = 1024$ fixed, varying number of random features from $m = 10$ to $m = 1800$.