A Double-Variational Bayesian Framework in Random Fourier Features for Indefinite Kernels

Fanghui Liu, *Member, IEEE*, Xiaolin Huang, *Senior Member, IEEE*, Lei Shi, Jie Yang, and Johan A. K. Suykens, *Fellow, IEEE*

***Abstract*— Random Fourier features (RFFs) have been successfully employed to kernel approximation in large-scale situations. The rationale behind RFF relies on Bochner's theorem, but the condition is too strict and excludes many widely used kernels, e.g., dot-product kernels (violates the shift-invariant condition) and indefinite kernels [violates the positive definite (PD) condition]. In this article, we present a unified RFF framework for indefinite kernel approximation in the reproducing kernel Kreĭn spaces (RKKSs). Besides, our model is also suited to approximate a dot-product kernel on the unit sphere, as it can be transformed into a shift-invariant but indefinite kernel. By the Kolmogorov decomposition scheme, an indefinite kernel in RKKS can be decomposed into the difference of two unknown PD kernels. The spectral distribution of each underlying PD kernel can be formulated as a nonparametric Bayesian Gaussian mixtures model. Based on this, we propose a double-infinite Gaussian mixture model in RFF by placing the Dirichlet process prior. It takes full advantage of high flexibility on the number of components and has the capability of approximating indefinite kernels on a wide scale. In model inference, we develop a non-conjugate variational algorithm with a sub-sampling scheme for the posterior inference. It allows for the non-conjugate case in our model and is quite efficient due to the sub-sampling strategy. Experimental results on several large classification data sets demonstrate the effectiveness of our nonparametric Bayesian model for indefinite kernel approximation when compared to other representative random feature-based methods.**

***Index Terms*— Indefinite kernel, kernel approximation, random Fourier features (RFFs), variational inference.**

Manuscript received December 1, 2018; revised April 23, 2019 and July 12, 2019; accepted August 2, 2019. Date of publication September 10, 2019; date of current version August 4, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61572315, Grant 61876107, Grant 61603248, Grant 11571078, Grant 11631015, and Grant U1803261, in part by the 973 Plan, China, under Grant 2015CB856004, in part by the Program of Shanghai Subject Chief Scientist under Grant 18XD1400700, in part by the European Research Council (ERC) Advanced Grant E-DUALITY under Grant 787960, and in part by KU Leuven under Grant CoE PFV/10/002 and Grant FWO G0A4917N. *(Corresponding authors: Xiaolin Huang; Jie Yang.)*

F. Liu is with the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai 200240, China, and also with the Department of Electrical Engineering (ESAT-STADIUS), KU Leuven, B-3001 Leuven, Belgium (e-mail: lfhsgre@outlook.com).

X. Huang and J. Yang are with the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai 200240, China, and also with the Institute of Medical Robotics, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: xiaolinhuang@sjtu.edu.cn; jieyang@sjtu.edu.cn).

L. Shi is with the Shanghai Key Laboratory for Contemporary Applied Mathematics, Fudan University, Shanghai 200433, China, and also with the School of Mathematical Sciences, Fudan University, Shanghai 200433, China (e-mail: leishi@fudan.edu.cn).

J. A. K. Suykens is with the Department of Electrical Engineering (ESAT-STADIUS), KU Leuven, B-3001 Leuven, Belgium (e-mail: johan.suykens@esat.kuleuven.be).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2019.2934729

I. Introduction

KERNEL methods [19]–[21] have enjoyed tremendous success in statistical machine learning with numerous applications, such as classification [22], regression [23], and dimensionality reduction [24], while a distinct bottleneck of kernel methods is their limited scalability in large data sets, i.e., the huge storage and significant computational cost of the kernel matrix. Given N observations, storing the kernel matrix often needs $\mathcal{O}(N^2)$ space and takes about $\mathcal{O}(N^2 d)$ operations, where d is the dimension. To make kernel methods scalable, kernel approximation is a powerful technique by mapping input features into a new space. With accurate kernel approximation, an efficient linear learner can be well trained in the transformed space while retaining the expressive power of nonlinear methods.

To overcome poor scaling in N, several routes have been explored. On the one hand, a straightforward way is employing the divide-and-conquer approach [12], [13]. It decomposes the full problem into several smaller easy-to-solve subproblems to accelerate the solving process. On the other hand, random projections are widely applicable and commonly used tactics to seek for a low-rank approximation, either data-dependent or data-independent. The data-dependent approaches approximate the kernel matrix by the greedy basis selection techniques [14], the incomplete Cholesky decomposition [15], or the Nyström methods [16]. In data-independent techniques, the kernel function is directly approximated by an explicit map, which is sampled from a distribution independent of training data. Most approaches that follow this idea are based on the random Fourier features (RFFs) [17] and have attracted significant attention to scale up kernel methods.

The theoretical foundation behind RFF is demonstrated by Bochner's theorem [30], i.e., any bounded, continuous, shift-invariant, and positive definite (PD) function can be expressed as the Fourier transform of a non-negative measure $\rho(\boldsymbol{w})$. However, Bochner's theorem requires the kernel to exhibit two properties: 1) shift-invariance, i.e., $\mathcal{K}(\boldsymbol{x}, \boldsymbol{y}) = \mathcal{K}(\boldsymbol{x} - \boldsymbol{y})$ and 2) positive definiteness. These two conditions exclude many widely used kernels, such as dot-product kernels and indefinite

kernels (real, symmetric, but not PD) [31]–[33]. For instance, the polynomial kernel and the Hellinger's kernel [34] are two dot-product kernels that do not satisfy the shift-invariant condition. Indefinite kernels include the hyperbolic tangent kernel [35], the TL1 kernel [36], and the Gaussian kernels with a geodesic distance on the manifold [37]. Moreover, dot-product kernels are commonly used on ℓ_2-normalized data to avoid the unboundedness [2], [38], so they can be reformulated as shift-invariant but not always PD on the unit sphere, i.e., $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = 1 - 0.5\|\boldsymbol{x} - \boldsymbol{y}\|^2$.

Pennington *et al.* [3] theoretically demonstrate that the Fourier transform of a polynomial kernel on the unit sphere is not a non-negative function, which is the obstruction to RFF. They empirically use ten Gaussians to approximate the polynomial kernel with spherical random features and employ a grid-search scheme for parameter tuning. This way is similar to a Gaussian mixture model (GMM) [39] for density estimation, but we need to carefully consider the following two issues. First, the number of Gaussian components is usually *ad hoc* pre-defined or manually specified. This is a real limitation as it significantly affects the approximation performance of indefinite kernels. Actually, it is difficult to argue that the number of Gaussian mixtures eventually runs up against some finite bound and remains fixed. We expect to infer the number of Gaussian components needed from data instead of the ungrounded guesses. Second, there are numerous parameters in GMM to be estimated, including the mixture coefficients, the mean vector, the covariance of each Gaussian model, and the number of Gaussian components. An efficient parameter estimation technique without heuristic pruning should be developed for model inference, especially when more Gaussians are taken into consideration.

In this article, we propose a fully non-parametric Bayesian model for approximating non-Bochner kernels (including the dot-product kernel on the unit sphere and shift-invariant indefinite kernel). In our framework, by the Kolmogorov decomposition scheme, an indefinite kernel in the reproducing kernel Kreĭn spaces (RKKSs) [31], [40] can be decomposed into the difference of two unknown PD kernels. The spectral distribution of each underlying PD kernel is modeled by an infinite GMM, resulting in a double-infinite GMM in RFF, termed as RFF-DIGMM. To be specific, our model treats the random frequency $\boldsymbol{w}$ as a latent parameter for each underlying PD kernel and places a Dirichlet process (DP) prior on it. This makes our random feature-based framework flexible to the indefinite kernel approximation. In model inference, we develop a non-conjugate variational inference method to infer the posterior distribution due to the non-conjugate random frequency $\boldsymbol{w}$ in RFF-DIGMM model. Furthermore, a sub-sampling scheme is used to accelerate the inference process.

Formally, the contributions are summarized as follows.

1) In light of the Kolmogorov decomposition scheme, we propose a double-infinite GMM for shift-invariant indefinite kernel approximation via random features. As a non-parametric Bayesian model, our model takes full advantage of high flexibility on the number of components and has the capability of approximating indefinite kernels in RKKS on a wide scale.
2) In the proposed RFF-DIGMM model, we design a non-conjugate variational inference algorithm with a sub-sampling scheme to infer the non-conjugate posterior distribution. The developed inference algorithm is feasible and efficient to accelerate the inference process for our non-conjugate model.
3) Experimental results illustrate that our RFF-DIGMM model is flexible to approximate indefinite kernels on a wide scale. Furthermore, its application to classification tasks on several large data sets demonstrates the superiority of our RFF-DIGMM model when compared to other representative random feature mapping-based algorithms.

The remainder of this article is organized as follows. Section II briefly introduces the preliminaries of RFFs and the stick-breaking construction for DP. Section III presents the proposed RFF-DIGMM model. The non-conjugate variational inference algorithm is given in Section IV. Section V shows the evaluation results of the proposed RFF-DIGMM model with other representative methods on several popular benchmarks. Finally, the conclusion is drawn in Section VI.

II. Preliminaries

This section briefly introduces the rationale of RFFs [17], [41] and stick-breaking construction for DP [42], [43]. Reviewing these two approaches will help to understand our double-infinite Gaussian mixtures model in RFF. Let $\mathcal{D} = \{\boldsymbol{x}_n\}_{n=1}^N$ be the sample set with N training examples with $\boldsymbol{x}_n \in \mathcal{X} \subseteq \mathbb{R}^d$. Let $\mathcal{K}(\cdot, \cdot)$ be a PD kernel function endowed in the reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ and $\boldsymbol{K} = [\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{N\times N}$ be the kernel matrix sampled from $\mathcal{D}$. The theoretical foundation of RFF relies on Bochner's celebrated characterization of PD functions.

Theorem 1 (Bochner's Theorem [30]): A continuous and shift-invariant function $\mathcal{K} : \mathbb{R}^d \to \mathbb{R}$ is PD if and only if it is the Fourier transform of a finite nonnegative Borel measure $\rho(\boldsymbol{w})$ on $\mathbb{R}^d$.

A consequence of Bochner's theorem is that any shift-invariant and PD kernel can be interpreted by

$$\begin{aligned}\mathcal{K}(\boldsymbol{x} - \boldsymbol{y}) &= \int_{\mathbb{R}^d} p(\boldsymbol{w}) \exp(\mathrm{i}\boldsymbol{w}^\top(\boldsymbol{x} - \boldsymbol{y}))\mathrm{d}\boldsymbol{w} \\ &= \mathbb{E}_{\boldsymbol{w}\sim\rho(\boldsymbol{w})}[\exp(\mathrm{i}\boldsymbol{w}^\top\boldsymbol{x}) \exp(\mathrm{i}\boldsymbol{w}^\top\boldsymbol{y})^*] \end{aligned} \tag{1}$$

where the symbol $\boldsymbol{x}^*$ denotes the complex conjugate of $\boldsymbol{x}$ and $\rho(\boldsymbol{w})$ can be scaled to a normalized density by setting $\mathcal{K}(0) = 1$. By the Monte Carlo integration, the kernel $\mathcal{K}$ can be approximated by

$$\mathcal{K}(\boldsymbol{x} - \boldsymbol{y}) \approx \frac{1}{M} \sum_{m=1}^{M} \exp\left(\mathrm{i}\boldsymbol{w}_m^\top\boldsymbol{x}\right) \exp\left(\mathrm{i}\boldsymbol{w}_m^\top\boldsymbol{y}\right)^* \tag{2}$$

where $\boldsymbol{w}_m$ is sampled i.i.d. from $\mathcal{P}$ with the density $\rho(\boldsymbol{w})$. In particular, since the kernel $\mathcal{K}$ is real-valued in most cases,

the imaginary part of (2) can be discarded, that is

$$\begin{aligned}&\mathcal{K}(\boldsymbol{x}-\boldsymbol{y})\\&\approx \varphi^{\top}(\boldsymbol{x})\varphi(\boldsymbol{y}), \text{ with } \varphi(\boldsymbol{x})\\&\triangleq \frac{1}{\sqrt{M}}\left[\cos\left(\boldsymbol{w}_1^{\top}\boldsymbol{x}\right),\ldots,\cos\left(\boldsymbol{w}_M^{\top}\boldsymbol{x}\right),\sin\left(\boldsymbol{w}_1^{\top}\boldsymbol{x}\right),\ldots,\sin\left(\boldsymbol{w}_M^{\top}\boldsymbol{x}\right)\right]^{\top}\end{aligned} \tag{3}$$

where $\varphi(\cdot) : \mathbb{R}^d \mapsto \mathbb{R}^M$ is the random feature mapping and $\varphi^{\top}(\boldsymbol{x})\varphi(\boldsymbol{y})$ is the unbiased estimation of $\mathcal{K}(\boldsymbol{x}, \boldsymbol{y})$. Hence, by random features, the storage and computational complexity can be reduced to $\mathcal{O}(NM)$ and $\mathcal{O}(NMd)$, respectively. Recent works on random features aim to improve the approximation quality by the Quasi-Monte Carlo sampling [44], random orthogonal matrix [41], or decrease the time and space complexity by Fastfood [45], quadrature-based features [46]. However, these algorithms mainly focus on shift-invariant and PD kernels and cannot be directly applied to the non-Bochner kernels. Only a few literature based on random features are able to deal with a polynomial kernel by the Maclaurin's approximation [1] and tensor sketching [2] or indefinite kernel approximation by the finite Gaussian mixtures [3].

Next, we briefly review the stick-breaking construction for DP [47]. DP is a stochastic process over discrete probability measures, i.e., atoms, with countably infinite support. It is widely used in the Bayesian nonparametric models of data, particularly in DP mixture models [48]. Mathematically, let G be a distribution over the probability space Θ, α be a positive real scalar, and H be a base measure over Θ. If any r partitions $(A_1, A_2, \ldots, A_r)$ of the corresponding probability space obey a Dirichlet distribution, then the distribution $(G(A_1), G(A_2), \ldots, G(A_r))$ is a DP

$$\begin{aligned}&(G(A_1),G(A_2),\ldots, G(A_r))\\&\qquad\sim \text{Dir}(\alpha H(A_1),\alpha H(A_2),\ldots,\alpha H(A_r))\end{aligned}$$

where r is a natural number [43] and α is the concentration parameter. We denote it as $G \sim \mathcal{DP}(\alpha, H)$.

To build a DP, one representative strategy is stick-breaking construction [8]. Given a unit-length stick (0, 1), we first draw $\beta_1 \sim \text{Beta}(1, \alpha_0)$, set $\theta_1 \triangleq \beta_1$, and pick the fraction $1-\beta_1$ as the remainder of the stick, and then, we draw $\beta_2 \sim \text{Beta}(1, \alpha_0)$ and assign $\theta_2 \triangleq \beta_2(1-\beta_1)$. Repeating this procedure, we have DP mixtures with stick-braking representation, i.e., the random measure G is associated with a DP $\mathcal{DP}(G_0, \alpha_0)$ with respect to the base distribution G_0 and the concentration parameter α_0. Mathematically, the construction can be formulated as

$$G = \sum_{k=1}^{\infty}\theta_k(\boldsymbol{\beta})\delta_{\Phi_k}, \quad \theta_k(\boldsymbol{\beta}) = \beta_k\prod_{s=1}^{k-1}(1-\beta_s) \tag{4}$$

with $\Phi_k \sim G_0$ and $\beta_k|\alpha_0 \sim \text{Beta}(1, \alpha_0)$. The notation δ_{Φ_k} is the Kronecker delta function, of which the value is 1 at location Φ_k and 0 elsewhere. It can be found that G is discrete almost surely, i.e., the support of G consists of a countably infinite set of atoms that are drawn independently of G_0. Since the distributions sampled from a DP are discrete almost surely, data generated from a DP mixture can be partitioned into different groups according to the distinct values of the sampled distributions. As a result, the whole model serves as a mixture model, in which the number of components is random and grows as new data are observed. For more details on the nonparametric Bayesian model and its construction, we refer the reader to [43] and [49].

III. Model Description

In this section, we present the formulation of our RFF-DIGMM model and its graphical model representation.

A. Kolmogorov Decomposition for Indefinite Kernels

In theory, a functional space spanned by indefinite kernels does not belong to the RKHSs [19], [50]. To investigate the indefinite kernels, we need Kreĭn spaces defined as follows.

Definition 1 (Kreĭn Space [40]): An inner product space is a Kreĭn space $\mathcal{H}_{\mathcal{K}}$ if there exist two Hilbert spaces $\mathcal{H}^+$ and $\mathcal{H}^-$, such that the following holds.

1) All $f \in \mathcal{H}_{\mathcal{K}}$ can be decomposed into $f = f^+ + f^-$, where $f^+ \in \mathcal{H}^+$ and $f^- \in \mathcal{H}^-$, respectively.
2) $\forall f, g \in \mathcal{H}_{\mathcal{K}}$, $\langle f, g\rangle_{\mathcal{H}_{\mathcal{K}}} = \langle f^+, g^+\rangle_{\mathcal{H}_+} - \langle f^-, g^-\rangle_{\mathcal{H}_-}$.

If $\mathcal{H}_+$ and $\mathcal{H}_-$ are two RKHSs, the Kreĭn space $\mathcal{H}_{\mathcal{K}}$ is an RKKS associated with a unique indefinite reproducing kernel $\mathcal{K}$, such that the reproducing property holds, i.e., $\forall f \in \mathcal{H}_{\mathcal{K}}$, $f(\boldsymbol{x}) = \langle f, k(\boldsymbol{x}, \cdot)\rangle_{\mathcal{H}_{\mathcal{K}}}$. To link indefinite kernels of RKKS to RKHS, we present a useful proposition as follows.

Proposition 1 [51, Proposition 2.1]: An indefinite reproducing kernel $\mathcal{K}$ associated with an RKKS admits a Kolmogorov decomposition

$$\mathcal{K} = \mathcal{K}^+ - \mathcal{K}^-$$

with two PD kernels $\mathcal{K}^+$ and $\mathcal{K}^-$.

Typical examples of indefinite kernels that admit the Kolmogorov decomposition include a wide range of commonly used indefinite kernels, such as a linear combination of PD kernels and conditional PD kernels. Hence, approximating an indefinite kernel $\mathcal{K}$ in RKKS by random features can be formulated as conducting random feature mappings for two underlying PD kernels $\mathcal{K}^+$ and $\mathcal{K}^-$.

Although the above-mentioned proposition presents the existence of a Kolmogorov decomposition for an indefinite kernel in RKKS, it does not provide a specific decomposition result for $\mathcal{K}^+$ and $\mathcal{K}^-$. In this case, what we only have is the indefinite kernel $\mathcal{K}$ and its associated indefinite kernel matrix $\boldsymbol{K}$ on the sample set $\mathcal{D}$. An intuitive way is to conduct an eigenvalue decomposition for $\boldsymbol{K}$, i.e., $\boldsymbol{K} = \boldsymbol{U}^{\top}\boldsymbol{\Gamma}\boldsymbol{U}$, where $\boldsymbol{U}$ is an orthogonal matrix and the diagonal matrix is $\boldsymbol{\Gamma} = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_N)$ with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq 0 \geq \cdots \geq \lambda_N$. Without loss of generality, we assume that the first s eigenvalues are nonnegative and the remaining $N-s$ ones are negative. Hence, $\boldsymbol{K}$ can be decomposed as $\boldsymbol{K} = \boldsymbol{K}^+ - \boldsymbol{K}^-$ with the following formulation:

$$\begin{cases}\boldsymbol{K}^+ = \boldsymbol{U}^{\top}\text{diag}(\lambda_1 + \tau, \ldots, \lambda_s + \tau)\boldsymbol{U}\\ \boldsymbol{K}^- = \boldsymbol{U}^{\top}\text{diag}(\tau - \mu_{N-s+1}, \ldots, \tau - \mu_N)\boldsymbol{U}\end{cases} \tag{5}$$

where τ is to ensure that these two matrices $\boldsymbol{K}^+$ and $\boldsymbol{K}^-$ are PD. Obviously, the decomposition for $\boldsymbol{K}^+$ and $\boldsymbol{K}^-$ is

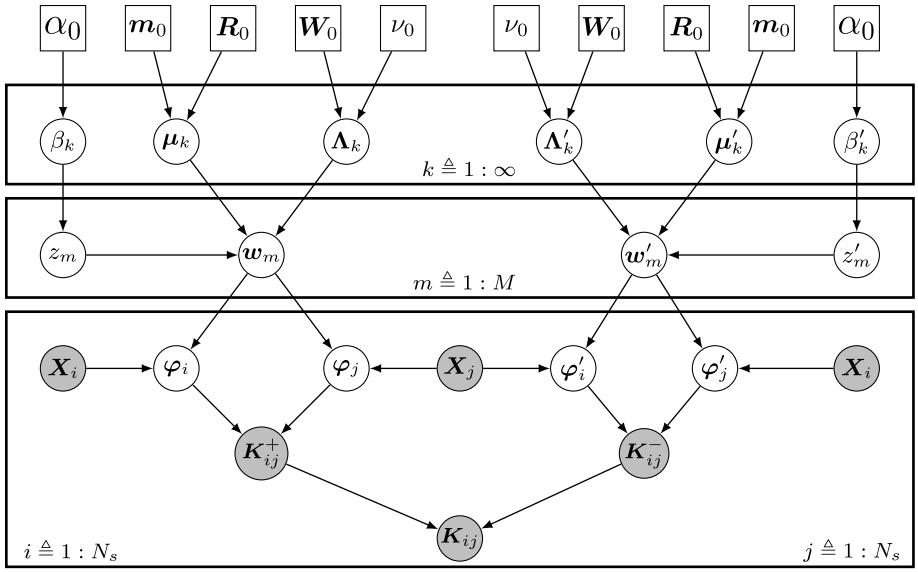


Fig. 1. Graphical model representation of RFF-DIGMM.

not unique due to the different choices of τ, and we will experimentally verify the influence of different eigenvalue decompositions in Section V-E. Furthermore, to speed up the computational efficiency in large-scale situations, we only consider a subset of training examples to conduct eigenvalue decomposition. That is, given two sub-matrices from $\boldsymbol{K}^+$ and $\boldsymbol{K}^-$, our target is to obtain random feature mappings for $\mathcal{K}^+$ and $\mathcal{K}^-$ by the proposed RFF-DIGMM model.

B. Graphical Model Representation for RFF-DIGMM

Bochner's theorem shows that the characteristic function (i.e., the inverse Fourier transformation) of a continuous distribution $\mathcal{P}$ with its pdf $\rho(\boldsymbol{w})$ is associated with a shift-invariant and PD kernel [52]. For example, suppose that $\rho(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is a Gaussian distribution with the mean vector $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$, its characteristic function is a shift-invariant kernel $\mathcal{K}(\Delta) = \exp(\mathrm{i}\boldsymbol{\mu}^\top\Delta - \frac{1}{2}\Delta^\top\boldsymbol{\Sigma}\Delta)$ with $\Delta := \boldsymbol{x} - \boldsymbol{y}$. That is to say, a Gaussian distribution and its characteristic function define a Gaussian kernel. Considering that GMM is a universal approximator for any continuous distribution [53] in density estimation, the spectral distribution $\mathcal{P}$ can be well approximated by GMM. From the kernel learning perspective, this mixture modeling is able to yield a general PD kernel, which provides a justification to obtain random feature mappings for $\mathcal{K}^+$ and $\mathcal{K}^-$, respectively.

For the underlying PD kernel $\mathcal{K}^+$, since the number of its corresponding nonnegative Borel measure $\rho(\boldsymbol{w})$ is not a prior known, we posit it as infinite, namely

$$\rho(\boldsymbol{w}) = \sum_{k=1}^{\infty} \theta_k \mathcal{N}\left(\boldsymbol{w}|\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1}\right) \tag{6}$$

where $\boldsymbol{\mu}_k$ and $\boldsymbol{\Lambda}_k$ are the mean vector and precision matrix of each Gaussian, respectively. According to Plancherel's theorem [18], the expression of $\rho(\boldsymbol{w})$ with infinite components in (6) is able to approximate any shift-invariant PD kernel. It relates spectral accuracies to the original domain by the following characteristic function:

$$\mathcal{K}^+(\boldsymbol{x}-\boldsymbol{y}) = \sum_{k=1}^{\infty} \theta_k \exp\left(\mathrm{i}\boldsymbol{\mu}_k^\top(\boldsymbol{x}-\boldsymbol{y}) - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{y})^\top\boldsymbol{\Lambda}_k(\boldsymbol{x}-\boldsymbol{y})\right).$$

In practical use, the kernel is often real-valued, so we consider the real part of the above-presented equation

$$\mathcal{K}^+(\Delta) = \sum_{k=1}^{\infty} \theta_k \exp\left(-\frac{1}{2}\Delta^\top\boldsymbol{\Lambda}_k\Delta\right)\cos\left(\boldsymbol{\mu}_k^\top\Delta\right)$$

with $\Delta := \boldsymbol{x} - \boldsymbol{y}$. In this case, the PD kernel $\mathcal{K}^+$ can be approximated by $\mathcal{K}^+(\boldsymbol{x}, \boldsymbol{y}) \approx \varphi^\top(\boldsymbol{x})\varphi(\boldsymbol{y})$, where $\varphi(\boldsymbol{x})$ is as in (3).

Likewise, for $\mathcal{K}^-$, its corresponding nonnegative Borel measure $\rho'(\boldsymbol{w}')$ is formulated as

$$\rho'(\boldsymbol{w}') = \sum_{k=1}^{\infty} \theta_k \mathcal{N}\left(\boldsymbol{w}'|\boldsymbol{\mu}'_k, \boldsymbol{\Lambda}'^{-1}_k\right) \tag{7}$$

and its characteristic function is

$$\mathcal{K}^-(\boldsymbol{x}-\boldsymbol{y}) = \sum_{k=1}^{\infty} \theta'_k \exp\left(\mathrm{i}\boldsymbol{\mu}_k'^\top(\boldsymbol{x}-\boldsymbol{y}) - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{y})^\top\boldsymbol{\Lambda}'_k(\boldsymbol{x}-\boldsymbol{y})\right).$$

In this case, the PD kernel $\mathcal{K}^-$ can be approximated by

$$\begin{aligned}
&\mathcal{K}^-(\boldsymbol{x}-\boldsymbol{y}) \\
&\approx \varphi'^\top(\boldsymbol{x})\varphi'(\boldsymbol{y}), \text{ with } \varphi'(\boldsymbol{x}) \\
&\triangleq \frac{1}{\sqrt{M}}\left[\cos\left(\boldsymbol{w}_1'^\top\boldsymbol{x}\right),\ldots,\cos\left(\boldsymbol{w}_M'^\top\boldsymbol{x}\right),\sin\left(\boldsymbol{w}_1'^\top\boldsymbol{x}\right),\ldots,\sin\left(\boldsymbol{w}_M'^\top\boldsymbol{x}\right)\right]^\top.
\end{aligned} \tag{8}$$

Therefore, the expression of $\rho(\boldsymbol{w})$ in (6) and $\rho'(\boldsymbol{w}')$ in (7) with infinite components provide adequate flexibility to find a good approximation of $\mathcal{K}$ from a broad class.

The graphical model representation of our RFF-DIGMM model is shown in Fig. 1. In our model, to speed up the

computational efficiency and to reduce the memory storage, we randomly select N_s examples from the training set $\mathcal{D}$, resulting in the sketch $\mathcal{D}_s$. Similar to [9] and [10], our model works between sub-sampling the training set and adjusting the hidden structure for parameter estimation based on the sketch $\mathcal{D}_s$. Thereby, finding a good approximation to a non-Bochner kernel over N_s observations can be represented as

$$K_{ij} = K_{ij}^{+} - K_{ij}^{-} = \varphi^{\top}(\boldsymbol{x}_i)\varphi(\boldsymbol{x}_j) - \varphi'^{\top}(\boldsymbol{x}_i)\varphi'(\boldsymbol{x}_j) + \epsilon, \quad \forall \boldsymbol{x}_i, \quad \boldsymbol{x}_j \in \mathcal{D}_s \text{ and } i \neq j \tag{9}$$

with $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$. The two random feature mappings φ and φ' satisfy $\mathcal{K}^{+}(\boldsymbol{x}_i - \boldsymbol{x}_j) \approx \varphi^{\top}(\boldsymbol{x}_i)\varphi(\boldsymbol{x}_j)$ and $\mathcal{K}^{-}(\boldsymbol{x}_i - \boldsymbol{x}_j) \approx \varphi'^{\top}(\boldsymbol{x}_i)\varphi'(\boldsymbol{x}_j)$, respectively. It is important to point out that, on each trial, we randomly sample two examples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ $(i \neq j)$ without replacement from $\mathcal{D}_s$ to construct K_{ij} and directly set $K_{ii} = 1$. By doing so, we are able to avoid the situation when the pair example $(\boldsymbol{x}_i, \boldsymbol{x}_j, K_{ij})$ is not mutually pairwise independent [54].

In our RFF-DIGMM model, since the explicit feature mappings φ and φ' in (9) are determined by $\rho(\boldsymbol{w})$ and $\rho'(\boldsymbol{w}')$, respectively, the distributions of K_{ij}^{+} and K_{ij}^{+} are

$$p\left(K_{ij}^{+}|(\boldsymbol{x}_i, \boldsymbol{x}_j), \varphi\right) \sim \mathcal{N}\left(\frac{1}{M}\sum_{m=1}^{M}\cos\left(\boldsymbol{w}_m^{\top}(\boldsymbol{x}_i - \boldsymbol{x}_j)\right), \sigma_\epsilon^2\right)$$
$$p\left(K_{ij}^{-}|(\boldsymbol{x}_i, \boldsymbol{x}_j), \varphi'\right) \sim \mathcal{N}\left(\frac{1}{M}\sum_{m=1}^{M}\cos\left(\boldsymbol{w}_m'^{\top}(\boldsymbol{x}_i - \boldsymbol{x}_j)\right), \sigma_\epsilon^2\right).$$

The random frequencies $\boldsymbol{w}_m$ and $\boldsymbol{w}_m'$ over the input space for a mixture component are given by

$$\begin{cases} p(\boldsymbol{w}_m|z_m = k, \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) \sim \mathcal{N}\left(\boldsymbol{w}_m|\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1}\right) \\ p(\boldsymbol{w}_m'|z_m' = k, \boldsymbol{\mu}_k', \boldsymbol{\Lambda}_k') \sim \mathcal{N}\left(\boldsymbol{w}_m'|\boldsymbol{\mu}_k', \boldsymbol{\Lambda}_k'^{-1}\right) \end{cases}$$

where z_m and z_m' are two latent variables that assign the indices of the parameter associated with $\boldsymbol{w}_m$ and $\boldsymbol{w}_m'$. The mean vectors $\boldsymbol{\mu}_k$ and $\boldsymbol{\mu}_k'$ and the precision matrices $\boldsymbol{\Lambda}_k$ and $\boldsymbol{\Lambda}_k'$ are further specified by the Gaussian distribution priors and the normal-Wishart distribution priors with the same hyperparameters, respectively. To be specific, the corresponding priors are

$$\boldsymbol{\mu}_k, \boldsymbol{\mu}_k' \sim \mathcal{N}\left(\boldsymbol{m}_0, \boldsymbol{R}_0^{-1}\right), \quad \boldsymbol{\Lambda}_k, \boldsymbol{\Lambda}_k' \sim \mathcal{W}(\boldsymbol{W}_0, \nu_0). \tag{10}$$

Besides, the distribution of z_m can be regarded as a multinomial distribution with parameters $\{\theta_k\}_{k=1}^{\infty}$ by the following formulation:

$$p(z_m|\beta_k) = \prod_{k=1}^{\infty}(1 - \beta_k)^{1[z_m > k]}\beta_k^{1[z_m = k]} \tag{11}$$

where β_k is given by (4), determining the mixing proportions $\{\theta_k\}_{k=1}^{\infty}$. The prior for $\{\theta_k\}_{k=1}^{\infty}$ is a DP prior built by the stick-breaking construction, so we define it by the stick-breaking distribution $\boldsymbol{\theta} \sim \text{GEM}(\alpha_0)$, where GEM (Griffiths–Engen–McCloskey) is the stick breaking prior [8]. The mixing proportions $\{\theta_k\}_{k=1}^{\infty}$ can be regarded as a sequence of sticks with lengths, satisfying $\sum_{k=1}^{\infty}\theta_k = 1$. The product $\prod_{s=1}^{k-1}(1 - \beta_s)$ denotes the previous remaining length of the stick, and multiplication by β_s gives the length of the stick currently broken off. Hence, (11) can be formulated as

$$z_m|\{\beta_1, \beta_2, \ldots, \beta_\infty\} \sim \text{Mult}(\boldsymbol{\beta})$$

where Mult denotes the multinomial distribution. Similarly, z_m' is subject to

$$p(z_m'|\beta_k') = \prod_{k=1}^{\infty}\left(1 - \beta_k'\right)^{1\left[z_m' > k\right]}\beta_k'^{1\left[z_m' = k\right]}.$$

Finally, the complete generative process is given as follows.

1) Draw the mixing proportions $\{\theta_i\}_{i=1}^{\infty} : \boldsymbol{\theta} \sim \text{GEM}(\alpha_0)$ and $\{\theta_i'\}_{i=1}^{\infty} : \boldsymbol{\theta}' \sim \text{GEM}(\alpha_0)$.
2) Draw the mixture components, for $k = 1 : \infty$
   a) Draw $\boldsymbol{\mu}_k, \boldsymbol{\mu}_k' \sim \mathcal{N}(\boldsymbol{m}_0, \boldsymbol{R}_0^{-1})$.
   b) Draw $\boldsymbol{\Lambda}_k, \boldsymbol{\Lambda}_k' \sim \mathcal{W}(\boldsymbol{W}_0, \nu_0)$.
3) For each random frequency index $m = 1, 2, \ldots, M$, carry out the following.
   a) Draw the indicate labels $z_m|\{\beta_1, \beta_2, \ldots, \beta_\infty\} \sim \text{Mult}(\boldsymbol{\theta}(\boldsymbol{\beta}))$ and $z_m'|\{\beta_1', \beta_2', \ldots, \beta_\infty'\} \sim \text{Mult}(\boldsymbol{\theta}'(\boldsymbol{\beta}'))$.
   b) Draw the random feature vectors $\boldsymbol{w}_m \sim \mathcal{N}(\boldsymbol{w}_m|z_m = k, \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1})$ and $\boldsymbol{w}_m' \sim \mathcal{N}(\boldsymbol{w}_m'|z_m' = k, \boldsymbol{\mu}_k', \boldsymbol{\Lambda}_k'^{-1})$.
4) For any two selected training examples $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{D}_s$, carry out the following.
   a) Compute $\varphi(\boldsymbol{x}_i)$, $\varphi(\boldsymbol{x}_j)$, $\varphi'(\boldsymbol{x}_i)$, and $\varphi'(\boldsymbol{x}_j)$ by (3).
   b) Draw an observation $K_{ij} \sim \mathcal{N}(\varphi^{\top}(\boldsymbol{x}_i)\varphi(\boldsymbol{x}_j) - \varphi'^{\top}(\boldsymbol{x}_i)\varphi'(\boldsymbol{x}_j), \sigma_\epsilon^2)$.

After conducting the generative process of our RFF-DIGMM model, we need to infer the associated parameters with respect to $\rho(\boldsymbol{w})$ and $\rho'(\boldsymbol{w}')$. For $\rho(\boldsymbol{w})$, defining the parameter sets $\tilde{\boldsymbol{\beta}} = \{\beta_1, \beta_2, \ldots, \beta_\infty\}$, $\tilde{\boldsymbol{\mu}} = \{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_\infty\}$, and $\tilde{\boldsymbol{\Lambda}} = \{\boldsymbol{\Lambda}_1, \boldsymbol{\Lambda}_2, \ldots, \boldsymbol{\Lambda}_\infty\}$ and the latent variable sets $\tilde{\boldsymbol{w}} = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_M\}$ and $\tilde{\boldsymbol{z}} = \{z_1, z_2, \ldots, z_M\}$, the hidden variable set is given by $\boldsymbol{\Omega} = \{\tilde{\boldsymbol{\beta}}, \tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\Lambda}}, \tilde{\boldsymbol{z}}, \tilde{\boldsymbol{w}}\}$. As illustrated by the graphical model shown in Fig. 1, the joint distribution of all the random variables with respect to ρ is given by

$$p(\mathcal{D}_s, \boldsymbol{\Omega}) = p(\tilde{\boldsymbol{\beta}})p(\tilde{\boldsymbol{\mu}})p(\tilde{\boldsymbol{\Lambda}})\prod_{m=1}^{M}p(z_m|\tilde{\boldsymbol{\beta}})p(\boldsymbol{w}_m|z_m, \tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\Lambda}}) \times \prod_{i,j=1, i\neq j}^{N_s}p\left(K_{ij}^{+}|(\boldsymbol{x}_i, \boldsymbol{x}_j), \tilde{\boldsymbol{w}}\right)$$

where the notations are $p(\tilde{\boldsymbol{\beta}}) = \prod_{k=1}^{\infty}p(\beta_k)$, $p(\tilde{\boldsymbol{\mu}}) = \prod_{k=1}^{\infty}p(\boldsymbol{\mu}_k)$, $p(\tilde{\boldsymbol{\Lambda}}) = \prod_{k=1}^{\infty}p(\boldsymbol{\Lambda}_k)$, $p(\tilde{\boldsymbol{z}}) = \prod_{m=1}^{M}p(z_m)$, and $p(\tilde{\boldsymbol{w}}) = \prod_{m=1}^{M}p(\boldsymbol{w}_m)$. Accordingly, we have

$$p(\mathcal{D}_s, \boldsymbol{\Omega}) = \prod_{k=1}^{\infty}p(\beta_k|\alpha_0)p(\boldsymbol{\mu}_k|\boldsymbol{m}_0, \boldsymbol{R}_0)p(\boldsymbol{\Lambda}_k|\boldsymbol{W}_0, \nu_0) \times \prod_{m=1}^{M}p(z_m|\tilde{\boldsymbol{\beta}})p(\boldsymbol{w}_m|z_m, \tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\Lambda}})\prod_{i,j=1, i\neq j}^{N_s}p\left(K_{ij}^{+}|(\boldsymbol{x}_i, \boldsymbol{x}_j), \tilde{\boldsymbol{w}}\right).$$

Likewise, the joint distribution of all the random variables with respect to ρ' is given by

$$p(\mathcal{D}_s, \Omega') = \prod_{k=1}^{\infty} p(\beta_k'|\alpha_0)\, p(\boldsymbol{\mu}_k'|\boldsymbol{m}_0, \boldsymbol{R}_0)\, p(\boldsymbol{\Lambda}_k'|\boldsymbol{W}_0, \nu_0) \times \prod_{m=1}^{M} p(z_m'|\tilde{\boldsymbol{\beta}}')\, p(\boldsymbol{w}_m'|z_m', \tilde{\boldsymbol{\mu}}', \tilde{\boldsymbol{\Lambda}}') \prod_{i,j=1, i\neq j}^{N_s} p(K_{ij}^{-}|(\boldsymbol{x}_i, \boldsymbol{x}_j), \tilde{\boldsymbol{w}}')$$

where the variable notations $\Omega' = \{\tilde{\boldsymbol{\beta}}', \tilde{\boldsymbol{\mu}}', \tilde{\boldsymbol{\Lambda}}', \tilde{\boldsymbol{z}}', \tilde{\boldsymbol{w}}'\}$ for ρ' share the similar formulation with the corresponding definitions for ρ. Since the posteriors $p(\Omega|\mathcal{D}_s)$ and $p(\Omega'|\mathcal{D}_s)$ are often intractable, in Section IV, we will approximate them using the mean-field variational inference.

IV. Inference

In this section, we develop a variant of the mean-field variational inference algorithm to tackle the non-conjugate variable $\boldsymbol{w}$ in our model. Here, we take $\rho(\boldsymbol{w})$ as an example to illustrate the inference process. The inference for $\rho'(\boldsymbol{w}')$ can be obtained in a similar way.

A. Truncated DP in Mean-Field Approach

Variational inference [42], [55] aims to find a distribution in a simple family that is close to the true posterior distribution $p(\Omega|\mathcal{D}_s)$ by a proxy $q(\Omega)$ with the following decomposition:

$$\ln p(\mathcal{D}_s) = \mathcal{L}(q) + \mathrm{KL}(q||p) \tag{12}$$

where the Kullback–Leibler (KL) divergence is defined as $\mathrm{KL}(q||p) = \int q(\Omega) \ln\{q(\Omega)/p(\Omega|\mathcal{D}_s)\}\mathrm{d}\Omega$ and $\mathcal{L}(q)$ is the lower bound of $\ln p(\mathcal{D}_s)$ with the expression $\mathcal{L}(q) = \int q(\Omega) \ln\{p(\mathcal{D}_s, \Omega)/q(\Omega)\}\mathrm{d}\Omega$. Variational inference can be formulated as minimizing the KL divergence from the variational distribution to the posterior distribution, which is equivalent to maximize the evidence lower bound (ELBO).

To formulate the variational posterior $q(\Omega)$, the posterior DP is approximated by a truncated stick-breaking representation [56]. That is, given a value T, we set $q(\beta_T = 1) = 1$ to guarantee that the mixture proportions θ_k are zero for $k > T$. Note that the variational distribution is truncated, but our model is a full DP and is not truncated. Based on the truncated DP, we adopt the mean-field approximation by the fully factorized variational distribution to approximate $p(\Omega|\mathcal{D}_s)$

$$q(\Omega|\mathcal{D}_s) = \prod_{t=1}^{T-1} q(\beta_t) \prod_{k=1}^{T} q(\boldsymbol{\mu}_k) q(\boldsymbol{\Lambda}_k) \prod_{m=1}^{M} q(\boldsymbol{w}_m) q(z_m)\,.$$

Using the above-mentioned full factorization formulation, we can solve $q(\Omega|\mathcal{D}_s)$ by maximizing the lower bound $\mathcal{L}(q)$ in (12). The logarithm of the optimized factor $q^*(\boldsymbol{\vartheta})$ with $\boldsymbol{\vartheta} \in \Omega$ is

$$\ln q^*(\boldsymbol{\vartheta}) = \mathbb{E}_{\Omega\setminus\boldsymbol{\vartheta}} \ln p(\mathcal{D}_s, \Omega) + \mathrm{const} \tag{13}$$

where $\mathbb{E}_{\Omega\setminus\boldsymbol{\vartheta}}$ is the expectation with respect to all other latent variables and "const" (short for c) denotes a constant that is independent of $\boldsymbol{\vartheta}$. Therefore, using the ELBO and the mean-field family, the posterior approximate is cast as an optimization problem. It can be efficiently solved by a coordinate ascent variational inference [7], and we detail this as follows.

B. Update Variational Factors

The optimization for each variational factor is conducted by the coordinate ascent variational inference. It iteratively optimizes each factor of the mean-field variational density while holding the others fixed, which climbs the ELBO to a local optimum. Here, we just state the results, and the derivations can be found in the Appendix.

1) $q(\beta_t)$: We absorb terms in (13) that are independent of β_t into the additive normalization constant and then get a Beta posterior approximating distribution

$$\beta_t \sim \mathrm{Beta}\left(1 + \sum_{m=1}^{M} q(z_m = t), \alpha_0 + \sum_{m=1}^{M} q(z_m > t)\right).$$

2) $q(z_m)$: Likewise, we do not consider irrelevant terms of z_m in (13). Defining $\Xi \triangleq \mathbb{E}_{\boldsymbol{w}_m, \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k}[(\boldsymbol{w}_m - \boldsymbol{\mu}_k)^\top \boldsymbol{\Lambda}_k (\boldsymbol{w}_m - \boldsymbol{\mu}_k)]$ and $\ln \hbar_{mk} \triangleq \mathbb{E}(\ln \beta_k) + \sum_{t=1}^{k-1} \mathbb{E}[\ln(1-\beta_t)] + (1/2)(\mathbb{E}\ln|\boldsymbol{\Lambda}_k| - d\ln(2\pi) - \Xi)$ and scaling $\tilde{\hbar}_{mk} = (\hbar_{mk}/\sum_{t=1}^{T} \hbar_{mt})$, we have $q(z_m = k) = \tilde{\hbar}_{mk}$. It means that z_m is chosen according to a multinomial probability distribution.

3) $q(\boldsymbol{\mu}_k)$: Keeping only the terms that have functional dependence on $\boldsymbol{\mu}_k$, we get a Gaussian posterior approximating distribution $\boldsymbol{\mu}_k \sim \mathcal{N}(\boldsymbol{\mu}_k|\boldsymbol{m}_k, \boldsymbol{R}_k^{-1})$ with the following mean vector and precision matrix:

$$\begin{cases} \boldsymbol{m}_k = \boldsymbol{R}_k^{-1}\left(\boldsymbol{R}_0\boldsymbol{m}_0 + \mathbb{E}(\boldsymbol{\Lambda}_k) \sum_{m=1}^{M} q(z_m = k)\mathbb{E}(\boldsymbol{w}_m)\right) \\ \boldsymbol{R}_k = \boldsymbol{R}_0 + \mathbb{E}(\boldsymbol{\Lambda}_k) \sum_{m=1}^{M} q(z_m = k)\,. \end{cases}$$

4) $q(\boldsymbol{\Lambda}_k)$: We only retain some terms with respect to $\boldsymbol{\Lambda}_k$ in (13), the approximating distribution is $\boldsymbol{\Lambda}_k \sim \mathcal{W}(\boldsymbol{\Lambda}_k|\boldsymbol{W}_k, \nu_k)$ with $\nu_k = \nu_0 + \sum_{m=1}^{M} q(z_m = k)$, and $\boldsymbol{W}_k^{-1}$ is formulated by

$$\boldsymbol{W}_k^{-1} = \boldsymbol{W}_0^{-1} + \sum_{m=1}^{M} q(z_m = k)\mathbb{E}(\boldsymbol{w}_m - \boldsymbol{\mu}_k)(\boldsymbol{w}_m - \boldsymbol{\mu}_k)^\top\,.$$

5) $q(\boldsymbol{w}_m)$: The equation for solving $\boldsymbol{w}_m$ is a little complex because $\boldsymbol{w}_m$ is involved in multiple variational factors. Due to the fact that $\boldsymbol{w}_m$ is a non-conjugate variable, here, we use the second-order Taylor expansion for the cosine function, i.e., $\cos[\boldsymbol{w}_m^\top(\boldsymbol{x}_i - \boldsymbol{x}_j)] \approx 1 - (1/2)\boldsymbol{w}_m^\top(\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \boldsymbol{w}_m$. Accordingly, we inspect (13) and read off those terms that involve $\boldsymbol{w}_m$. Defining

$$\boldsymbol{S} \triangleq \sum_{k=1}^{T}[q(z_m = k)\mathbb{E}(\boldsymbol{\Lambda}_k)] + \frac{1}{2\sigma_\epsilon^2} \sum_{i,j=1, i\neq j}^{N_s} \left(1 - K_{ij}^{+}\right)(\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top$$

we get the posterior approximating distribution for $\boldsymbol{w}_m$

$$\boldsymbol{w}_m \sim \mathcal{N}\left(\boldsymbol{S}^{-1}\left\{\sum_{k=1}^{T}[q(z_m=k)\mathbb{E}(\boldsymbol{\Lambda}_k)\mathbb{E}(\boldsymbol{\mu}_k)]\right\}, \boldsymbol{S}^{-1}\right).$$

The variational distribution $q(\boldsymbol{w}_m)$ is subject to a Gaussian distribution. Its Gaussian form naturally stems from the Taylor approximation of the cosine function. By Bochner's theorem, we have $\mathbb{E}[\cos(\boldsymbol{w}_m^\top \bar{\boldsymbol{x}})] = \exp(-\|\bar{\boldsymbol{x}}\|^2/2)$ with $\bar{\boldsymbol{x}} := (\boldsymbol{x}_i - \boldsymbol{x}_j/\sigma)$. Hence, with a proper scale width σ, we can guarantee that $\langle \boldsymbol{w}_m, \boldsymbol{x}_i - \boldsymbol{x}_j\rangle = 0$ with high probability when $\|\bar{\boldsymbol{x}}\|$ approaches to zero, and accordingly, the Taylor approximation condition is satisfied. This approximation technique can also be found in the Laplace approximation variational inference for non-conjugate models [11]. Unlike the Laplace approximation, our variational inference algorithm does not require the exponential family assumption but directly uses the Taylor approximation of the cosine function.

Finally, by repeating the above-mentioned update steps, we adjust the free variational parameters to approximate the original distribution $p(\boldsymbol{\Omega}|\mathcal{D}_s)$ until convergence. Likewise, the variational approximation for $p(\boldsymbol{\Omega}'|\mathcal{D}_s)$ can be obtained in a similar fashion. The variational inference algorithm for model inference is summarized in Algorithm 1. The convergence results of our model are similar to the Laplace approximation method in [11], which converges to a local optimum of the variational objective. Here, we assess convergence by measuring the difference between the two consecutive iterations for $q(\boldsymbol{z})$. This is a common stopping criterion, and we set the maximum iteration number IterMAX to 50. We will experimentally verify the convergence of the proposed inference algorithm in Section V-F.

Algorithm 1: Variational Inference for RFF-DIGMM

Construct $\mathcal{D}_s$ and the associated sub-kernel matrix $\boldsymbol{K}$.
Obtain $\boldsymbol{K}^+$ and $\boldsymbol{K}^-$ by eigenvalue decomposition for $\boldsymbol{K}$.
Set IterMAX= 50, iter = 0, initialize variational
distributions $q(\boldsymbol{\Omega}|\mathcal{D}_s)$ and $q(\boldsymbol{\Omega}'|\mathcal{D}_s)$.
Repeat
  iter = iter + 1;
  for $k = 1$ *to* T **do**
    Update $q(\beta_k)$, $q(\boldsymbol{\mu}_k)$, $q(\boldsymbol{\Lambda}_k)$, $q(\beta'_k)$, $q(\boldsymbol{\mu}'_k)$, $q(\boldsymbol{\Lambda}'_k)$;
  end
  for $m = 1$ *to* M **do**
    Update $q(z_m)$, $q(\boldsymbol{w}_m)$, $q(z'_m)$, and $q(\boldsymbol{w}'_m)$;
  end
Until $\|q(\boldsymbol{z}^{\text{iter}}) - q(\boldsymbol{z}^{\text{iter-1}})\|_{\text{F}} \leq 1e^{-5}$ *or* iter=IterMAX;
return variational distributions $q(\boldsymbol{\Omega}|\mathcal{D}_s)$, $q(\boldsymbol{\Omega}'|\mathcal{D}_s)$ and
random features $\{\boldsymbol{w}_m\}_{m=1}^M$, $\{\boldsymbol{w}'_m\}_{m=1}^M$.

1) Complexity: Our inference algorithm involves simple computations, such as matrix addition and matrix multiplication, except that inferring $\boldsymbol{w}$ and $\boldsymbol{\Lambda}$ needs to conduct $d \times d$ matrix inversion operations, leading to $\mathcal{O}((M+T)d^3)$. Owing to the sub-sampling scheme, based on $\mathcal{D}_s$, the total runtime per iteration is $\mathcal{O}((M+T)d^3 + MN_sT + MN_s)$. As a result, our method is quite efficient because the inference is independent of N, especially on large data sets with $N \gg d$.

2) Prediction: The main focus of our RFF-DIGMM model is not limited to improve the quality of kernel approximation. Instead, we aim to train a linear classifier in the feature space endowed by the obtained random features for classification tasks.

TABLE I
DATA SET STATISTICS

datasets	feature dimension d	#traing examples	#test examples
ijcnn1	22	49,990	91,701
covtype	54	464,810	116,202
skin	3	122,529	122,528
EEG	14	7,490	7,490
spambase	57	2,301	2,300

V. EXPERIMENTS

In this section, we experimentally evaluate the approximation performance of the proposed RFF-DIGMM model and apply it to classification tasks. All the experiments implemented in MATLAB are repeated over ten runs on a standard PC with Intel i5-6500 CPU (3.20 GHz) and 16-GB RAM. The source code of our implementation can be found in `http://www.lfhsgre.org`.

A. Experiment Setup

1) Data Sets: We extensively study the proposed method on five large classification benchmark data sets[1] that are listed in Table I. The data in these data sets are normalized to $[0, 1]^d$ in advance, and we randomly pick half of the data for training and the rest for test on skin, EEG, and spambase. For ijcnn1, both training and test data have been divided. Following [12], we use a random 80%-20% split on covtype. Besides, our method is also evaluated on the MNIST data set [58]. It is a 28×28 (the feature dimension is $d = 784$) grayscale handwritten digits database with 50 000 images for training and 10 000 for test.

2) Kernel Setting: Experiment results here are based on four non-Bochner kernels, including two dot-product kernels on the unit sphere and two indefinite kernels, as listed in Table II. These four non-Bochner kernels can be transformed into indefinite but shift-invariant kernels and approximated by our RFF-DIGMM model.

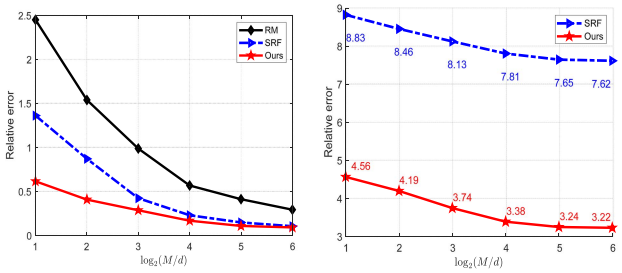
3) Parameter Setting: In our experiment, the sketch size is set to $N_s = 5$. The truncation parameter in DP is $T = 5$. The order in the polynomial kernel $\mathcal{K}_p(\boldsymbol{x}, \boldsymbol{y})$ is fixed with $p = 10$, and the parameters in the TL1 kernel and tanh kernel are set to $\tau = 0.7d$ and $\upsilon = 1/d$ as suggested.

4) Compared Methods: We choose the liblinear classifier [59] as our fast solver and present a comparison of our method (RFF-DIGMM) with the following algorithms.

[1]All the data sets can be downloaded from `https://www.csie.ntu.edu.tw/^cjlin/libsvmtools/datasets/` or the UCI Machine Learning Repository [57].

TABLE II
USED NON-BOCHNER KERNELS

Type	Kernel	Formulation
dot-product (sphere)	polynomial kernel	$\mathcal{K}_p(\boldsymbol{x}, \boldsymbol{y}) = (1 + \langle \boldsymbol{x}, \boldsymbol{y} \rangle)^p$
	Hellinger's kernel [34]	$\mathcal{K}_h(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{\langle \boldsymbol{x}, \boldsymbol{y} \rangle}$
indefinite	TL1 kernel [36]	$\mathcal{K}_\tau(\boldsymbol{x}, \boldsymbol{y}) = \max\{\tau - \|\boldsymbol{x} - \boldsymbol{y}\|_1, 0\}$
	tanh kernel [35]	$\mathcal{K}_v(\boldsymbol{x}, \boldsymbol{y}) = tanh(1 + v\langle \boldsymbol{x}, \boldsymbol{y} \rangle)$



(a) (b)

Fig. 2. Comparison of RMSE on EEG with (a) polynomial kernel and (b) TL1 kernel.

1) *Liblinear [59]:* It is an efficient solver for linear support vector machine (SVM). It serves as a baseline for comparison. The balance parameter C in liblinear is well tuned by fivefold cross validation on a grid of points: $C = [2^{-5}, 2^{-4}, \ldots, 2^5]$.
2) *RM [1]:* It adopts random Maclaurin feature maps to approximate the polynomial kernels but is infeasible to the Hellinger's kernel. This is because the Maclaurin expansion in RM requires the order of $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ not less than 1. Note that RM is not suited to the indefinite kernel as well.
3) *SRF [3]:* The polynomial kernel on the unit spherical is approximated by a GMM with ten components. Parameters in GMM are off-line optimized by the grid search over $[0, 2]$.

B. Quality of Kernel Approximation

One target of the experiment is to study the approximation quality of the non-Bochner kernels. In our experiment, we choose the polynomial kernel on the unit sphere and the TL1 kernel as examples. For them, we compute the ground-truth kernel matrix $\boldsymbol{K}^*$ and the approximated kernel matrix $\boldsymbol{K}$ on the EEG data set and validate the approximation quality of competing methods. The used evaluation metric here is the root mean square error (RMSE) between $\boldsymbol{K}^*$ and $\boldsymbol{K}$ over N observations, i.e., RMSE $= ((1/N(N-1)) \sum_{i=1}^{N} \sum_{j=1, j\neq i}^{N} (K_{ij}^* - K_{ij})^2)^{1/2}$.

Fig. 2 shows the kernel approximation performance of the compared algorithms with the polynomial kernel and the TL1 kernel on EEG. It can be observed that, in terms of polynomial kernel approximation, under varying random feature dimensionality, our method always provides less RMSE than RM and SRF, especially when using the lower dimensional random features. For the TL1 kernel approximation, along with the number of random features increases, the approximation error provided by SRF and our method steadily declines. Nevertheless, SRF yields a considerable approximation error and relatively large variance. Unlike SRF, our method achieves lower RMSE, which benefits from the high flexibility of the proposed RFF-DIGMM model. Notice that, the obtained RMSE on the TL1 kernel of both two methods is not as good as those for the polynomial kernel. This is mainly due to the non-smoothness of the TL1 kernel, which enhances the approximation difficulty.

C. Classification Results for Approximating Indefinite Kernels

The main focus of this article is to train a classifier by the obtained random features from RFF-DIGMM and then evaluate its classification accuracy on various data sets.

1) Classification Results on the UCI Database: For the polynomial kernel, we compare the performance of random feature mappings (RM, SRF, and our method) with the polynomial kernel and the liblinear method. For the Hellinger's kernel $\mathcal{K}_h(\boldsymbol{x}, \boldsymbol{y}) = (\langle \boldsymbol{x}, \boldsymbol{y} \rangle)^{1/2}$, SRF and our RFF-DIGMM method are taken into comparisons, but RM is not suited to this kernel. This is because the Maclaurin expansion in RM requires the order of $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ not less than 1. Table III reports the classification accuracy and the approximation time of all the competing methods for the polynomial kernel and Hellinger's kernel. As expected, the test accuracy improves with higher dimensional feature maps. The kernel approximation time linearly increases as the number of random features dimensionality raises. Among all the five data sets, our method achieves the best test accuracy. As a full Bayesian model, our RFF-DIGMM achieves comparable computational efficiency and accordingly decreases the computational cost for hyper-parameter tuning and multiple trials for determining a proper number of components by SRF.

Apart from the dot-product kernels, we also evaluate the classification performance of our model with the TL1 kernel and tanh kernel. The compared two algorithms include SRF and liblinear. The experimental results with respect to the test accuracy and training time are reported in Table IV. We can find that the proposed RFF-DIGMM model is superior to SRF in all the cases except for $M = 2d$ on ijcnn1 and spambase. For the TL1 kernel, the test accuracy of SRF is inferior to our method, and it almost stays unchanged with nearly indiscernible improvements on ijcnn1 and covtype even if M varies from $2d$ to $32d$. This phenomenon also appears to SRF with the tanh kernel on spambase. Instead, our RFF-DIGMM model flexibly exploits the infinite components that adapt to data and accordingly achieving promising test accuracy on these five data sets with varying M. The classification results on two indefinite kernels demonstrate the superiority of our RFF-DIGMM model.

2) Classification Results on Digit Recognition on MNIST: Apart from the above-mentioned five relatively low-dimensional data sets, here, we evaluate our RFF-DIGMM model on a relative high-dimensional data set, e.g., the MNIST data set [58]. Table V shows the recognition rates and time costs for kernel approximation of various compared algorithms

TABLE III

COMPARISON RESULTS OF VARIOUS ALGORITHMS WITH THE POLYNOMIAL KERNEL AND HELLINGER'S KERNEL FOR VARYING FEATURE MAP DIMENSIONALITY (M) IN TERMS OF CLASSIFICATION ACCURACY (MEAN ± STD. DEVIATION %) AND TRAINING TIME (MEAN ± STD. DEVIATION SEC.). THE BEST PERFORMANCE IS HIGHLIGHTED IN **BOLDFACE**

	Dataset	Method	$M = 2d$	$M = 8d$	$M = 16d$	$M = 32d$	liblinear
			Acc:% (time:sec.)	Acc:% (time:sec.)	Acc:% (time:sec.)	Acc:% (time:sec.)	Acc:%
Polynomial kernel	*ijcnn1*(d=22)	RM	**90.4**±0.4 (0.3±0.0)	93.3±0.3 (1.0±0.1)	95.1±0.7 (2.4±0.2)	96.5±0.4 (5.0±0.2)	92.5±0.0
		SRF[1]	81.2±1.9 (0.3±0.0)	86.6±0.7 (1.1±0.1)	88.8±0.7 (2.2±0.1)	93.4±0.5 (4.6±0.1)	
		Ours	90.0±0.6 (0.7±0.4)	**95.7**±0.4 (2.1±0.8)	**97.3**±0.4 (3.9±0.9)	**97.8**±0.1 (10.7±0.9)	
	covtype(d=54)	RM	72.9±0.6 (5.5±0.3)	75.9±0.3 (21.8±0.5)	77.8±0.1 (44.1±1.3)	79.3±0.3 (84.3±3.2)	75.6±0.2
		SRF	68.8±0.8 (3.7±0.1)	77.3±0.3 (16.2±0.5)	80.4±0.2 (49.5±5.4)	81.9±0.2 (93.6±8.4)	
		Ours	**73.7**±0.5 (1.7±0.6)	**79.5**±0.3 (3.0±1.1)	**80.9**±0.2 (6.0±0.4)	**83.1**±0.3 (19.8±2.2)	
	skin(d=3)	RM	80.9±6.8 (0.1±0.0)	87.9±8.2 (0.2±0.0)	87.5±7.3 (0.3±0.1)	87.7±7.7 (0.6±0.0)	91.1±0.1
		SRF	91.9±2.3 (0.1±0.0)	97.8±0.1 (0.3±0.0)	98.0±0.1 (0.4±0.0)	98.1±0.1 (0.9±0.0)	
		Ours	**98.0**±0.5 (0.5±0.2)	**98.1**±0.1 (0.7±0.3)	**98.2**±0.0 (1.3±0.6)	**98.2**±0.1 (3.1±1.3)	
	EEG(d=14)	RM	64.1±1.7 (0.0±0.0)	70.3±5.0 (0.1±0.0)	74.8±2.8 (0.1±0.0)	77.8±3.8 (0.1±0.0)	63.8±0.2
		SRF	66.3±0.8 (0.0±0.0)	67.9±0.7 (0.1±0.0)	71.8±1.8 (0.1±0.0)	74.4±2.1 (0.2±0.0)	
		Ours	**68.7**±1.2 (0.3±0.2)	**80.9**±0.5 (0.9±0.5)	**83.9**±0.5 (1.7±0.8)	**85.1**±0.5 (4.0±0.8)	
	spambase(d=57)	RM	**84.6**±1.4 (0.0±0.0)	**87.5**±2.2 (0.1±0.0)	87.1±3.3 (0.2±0.0)	90.2±0.5 (0.3±0.0)	90.7±0.8
		SRF	72.4±0.8 (0.0±0.0)	80.5±2.3 (0.1±0.0)	83.2±0.4 (0.2±0.0)	84.1±0.3 (0.4±0.0)	
		Ours	79.9±0.2 (0.5±0.2)	86.7±1.2 (1.7±0.3)	**88.4**±0.9 (3.8±0.2)	**90.9**±0.5 (6.2±0.2)	
Hellinger's kernel	*ijcnn1*(d=22)	SRF	87.1±0.8 (0.3±0.0)	91.4±0.6 (1.1±0.0)	94.3±0.4 (2.2±0.0)	96.7±0.2 (4.5±0.1)	92.5±0.0
		Ours	**89.5**±0.8 (1.2±0.3)	**95.0**±1.6 (5.8±1.5)	**97.0**±0.4 (10.4±3.2)	**97.8**±0.5 (27.6±6.7)	
	covtype(d=54)	SRF	72.0±0.7 (3.5±0.1)	78.7±0.1 (14.7±0.6)	80.1±0.3 (29.8±2.4)	82.4±0.5 (58.4±8.7)	75.6±0.2
		Ours	**74.6**±0.7 (1.9±0.6)	**79.7**±0.2 (5.5±0.5)	**81.3**±0.4 (12.5±0.4)	**83.0**±0.5 (27.5±2.7)	
	skin(d=3)	SRF	96.0±1.9 (0.1±0.0)	97.9±0.2 (0.2±0.0)	98.0±0.1 (0.5±0.0)	98.1±0.1 (0.9±0.0)	91.1±0.1
		Ours	**98.3**±0.3 (0.6±0.2)	**98.2**±0.1 (1.6±0.8)	**98.2**±0.0 (3.8±1.2)	**98.2**±0.1 (6.7±2.6)	
	EEG(d=14)	SRF	65.3±2.4 (0.0±0.0)	75.5±0.7 (0.1±0.0)	82.7±0.9 (0.2±0.0)	84.2±0.5 (0.4±0.1)	63.8±0.2
		Ours	**69.3**±1.6 (0.3±0.1)	**81.0**±1.3 (1.0±0.6)	**83.9**±0.7 (1.8±0.7)	**84.3**±0.8 (4.7±1.3)	
	spambase(d=57)	SRF	75.3±1.7(0.0±0.0)	78.4±1.3 (0.1±0.0)	81.2±0.5 (0.2±0.0)	83.3±1.1 (0.4±0.1)	90.7±0.8
		Ours	**78.4**±1.2 (1.5±0.4)	**84.6**±1.0 (3.9±1.5)	**87.6**±0.8 (9.2±3.2)	**88.2**±0.2 (12.9±4.1)	

[1] For each dataset, SRF obtains parameters in GMM by an off-line grid search scheme in advance, of which the time cost is reported as follows.

	ijcnn1	*covtype*	*skin*	*EEG*	*spambase*
the polynomial kernel (sec.)	16.7s	86.1s	18.1s	6.8s	41.3s
the Hellinger's kernel (sec.)	28.8s	21.9s	19.3s	16.2s	20.4s

with the polynomial kernel and the TL1 kernel. Since the feature dimension of this database is quite high, we just report the results under the setting of $M = 2d$. We can see that the proposed RFF-DIGMM model achieves the best performance on classification accuracy, which is narrowly followed by RM and SRF. In terms of time cost, our RFF-DIGMM model achieves an acceptable computational efficiency when compared to RM and SRF.

D. Compared With Other Kernel Approximation Methods With Bochner Kernels

As aforementioned, research works on approximating non-Bochner kernels by random features appear to be quite rare. Albeit this, we also compare the proposed RFF-DIGMM model with other recent kernel approximation algorithms as follows.

1) *RF [4]:* It is a nonparametric kernel learning framework by learning from optimal random features.
2) *Recursive-Nyström [5]:* It is a Nyström method based on recursive leverage score sampling.
3) *CROiclassification [6]:* A new concomitant rank order (CRO) kernel is proposed to approximate the Gaussian kernel on the unit sphere by random features.

The used kernel in Recursive-Nyström [5] and CROiclassification [6] is a Gaussian kernel. Instead, as a data-dependent method, RF considers the learned random features for kernel learning and approximation. In the current setting, our method, equipped with the polynomial kernel and $M = 32d$, is taken into consideration. For the subsequent classification, all of these three algorithms are combined with the liblinear algorithm for a fair comparison. The corresponding classification results are reported in Table VI. We find that RF appears to obtain a not very promising performance even if the kernel is learned instead of directly specified. Compared to [5] and [6] equipped with the Gaussian kernel, our method with the polynomial kernel achieves a comparable classification performance and computational cost. Actually, in this article, we do not want to claim that our RFF-DIGMM model is better than these two kernel approximation methods, as the scope of their applications is not the same. Instead, our aim is to show that the proposed RFF-DIGMM model provides a justification to conduct random features for non-Bochner kernels.

E. Parametric Analysis

Here, we study the influence of different sizes of the sketch, different truncation parameters, and different eigenvalue decompositions on the final results.

1) Size of the Sketch: In our RFF-DIGMM model, in each iteration, we sample N_s data points from $\mathcal{D}$ for variational

TABLE IV

COMPARISON RESULTS OF VARIOUS ALGORITHMS WITH THE TL1 KERNEL AND THE HYPERBOLIC TANGENT KERNEL FOR VARYING FEATURE MAP DIMENSIONALITY (M) IN TERMS OF CLASSIFICATION ACCURACY (MEAN ± STD. DEVIATION %) AND TRAINING TIME (MEAN ± STD. DEVIATION SEC.). THE BEST PERFORMANCE IS HIGHLIGHTED IN **BOLDFACE**

	Dataset	Method	$M = 2d$	$M = 8d$	$M = 16d$	$M = 32d$	liblinear
			Acc:% (time:sec.)	Acc:% (time:sec.)	Acc:% (time:sec.)	Acc:% (time:sec.)	Acc:%
TL1 kernel	*ijcnn1*(d=22)	SRF[1]	**91.0**±0.7 (0.3±0.0)	92.0±0.2 (1.0±0.0)	92.2±0.5 (2.0±0.1)	92.1±0.2 (4.3±0.4)	92.5±0.0
		Ours	89.8±0.5 (0.6±0.4)	**95.0**±0.4 (1.8±0.9)	**97.1**±0.3 (3.8±0.8)	**97.4**±0.3 (7.8±0.7)	
	covtype(d=54)	SRF	72.7±0.7 (3.3±0.1)	73.6±0.3 (16.0±0.5)	73.7±0.2 (27.7±2.1)	73.8±0.4 (41.8±14.6)	75.6±0.2
		Ours	**73.5**±0.6 (1.9±0.6)	**86.7**±0.4 (6.3±0.5)	**87.2**±0.2 (10.8±0.3)	**87.2**±0.6 (20.0±1.2)	
	skin(d=3)	SRF	95.4±1.7 (0.1±0.0)	97.9±0.2 (0.2±0.0)	98.1±0.2 (0.4±0.0)	98.1±0.1 (0.8±0.0)	91.1±0.1
		Ours	**98.0**±0.5 (0.3±0.2)	**98.1**±0.0 (0.7±0.3)	**98.1**±0.1 (1.8±0.6)	**98.2**±0.0 (2.4±0.7)	
	EEG(d=14)	SRF	66.8±2.6 (0.1±0.0)	77.7±0.9 (0.1±0.0)	84.6±1.0(0.2±0.0)	**89.8**±0.5 (0.4±0.0)	63.8±0.2
		Ours	**69.0**±1.0 (0.3±0.2)	**80.6**±1.0 (0.9±0.6)	**85.2**±0.3 (1.7±0.8)	86.9±0.8 (3.9±0.8)	
	spambase(d=57)	SRF	**84.1**±2.4(0.0±0.0)	86.0±1.1 (0.1±0.0)	87.0±0.7 (0.2±0.0)	88.3±1.0 (0.4±0.1)	90.7±0.8
		Ours	79.9±1.0 (0.9±0.6)	**86.7**±1.1 (2.8±1.0)	**88.0**±0.9 (5.5±1.0)	**88.9**±0.6 (9.8±0.9)	
tanh kernel	*ijcnn1*(d=22)	SRF	**92.1**±0.4 (0.3±0.0)	93.0±0.4 (1.1±0.0)	93.9±0.4 (2.2±0.1)	95.5±0.8 (4.2±0.2)	92.5±0.0
		Ours	90.1±0.3 (0.6±0.4)	**95.6**±0.4 (1.9±0.9)	**97.5**±0.4 (3.7±0.9)	**97.9**±0.2 (10.4±1.2)	
	covtype(d=54)	SRF	75.8±0.8 (3.3±0.1)	78.9±0.1 (13.8±0.6)	80.4±0.3 (28.7±2.8)	82.8±0.6 (44.7±12.1)	75.6±0.2
		Ours	**74.2**±1.2 (1.7±0.5)	**79.6**±0.1 (4.9±0.7)	**81.0**±0.2 (10.4±0.5)	**83.6**±0.5 (22.5±1.5)	
	skin(d=3)	SRF	94.2±2.3 (0.1±0.0)	**98.2**±0.2 (0.2±0.0)	**98.2**±0.1 (0.4±0.0)	98.1±0.1 (0.8±0.0)	91.1±0.1
		Ours	**97.2**±1.5 (0.6±0.2)	**98.2**±0.1 (1.4±0.5)	**98.2**±0.0 (2.1±0.8)	**98.2**±0.1 (4.2±0.8)	
	EEG(d=14)	SRF	64.3±1.2 (0.1±0.0)	75.2±1.1 (0.2±0.0)	78.4±0.8 (0.3±0.0)	79.1±0.9 (0.5±0.1)	63.8±0.2
		Ours	**69.3**±1.2 (0.3±0.2)	**80.2**±1.0 (1.9±0.8)	**83.4**±0.5 (1.7±0.8)	**84.5**±1.2 (4.0±0.7)	
	spambase(d=57)	SRF	**83.5**±1.8(0.1±0.0)	84.2±1.1 (0.1±0.0)	84.8±0.8 (0.2±0.0)	85.0±1.4 (0.4±0.1)	90.7±0.8
		Ours	76.2±1.4 (2.0±0.7)	**85.6**±1.5 (4.8±1.0)	**87.0**±0.8 (7.5±2.5)	**87.2**±1.2 (20.7±4.8)	

[1] For each dataset, SRF obtains parameters in GMM by an off-line grid search scheme in advance, of which the time cost is reported as follows.

	ijcnn1	*covtype*	*skin*	*EEG*	*spambase*
the TL1 kernel (sec.)	3.7s	8.3s	10.7s	3.0s	6.5s
the tanh kernel (sec.)	10.2s	11.8s	29.5s	16.5s	17.4s

TABLE V

COMPARISON RESULTS OF VARIOUS ALGORITHMS ON MNIST DATA SET. THE BEST SCORES ARE HIGHLIGHTED BY **BOLDFACE**

liblinear	polynomial kernel			TL1 kernel	
	RM	SRF	Ours	SRF	Ours
Acc:% (time:sec.)	Acc:% (time:sec.)	Acc:% (time:sec.)	Acc:% (time:sec.)	Acc:% (time:sec.)	Acc:% (time:sec.)
91.9±0.0 (39.7±0.6)	94.6±0.0 (87.0±0.8)	90.2±0.1 (129.8±4.6)	**94.9**±0.0 (171.8±3.2)	94.1±0.1 (75.8±0.2)	**95.2**±0.1 (144.3±2.7)

TABLE VI

COMPARISON RESULTS OF VARIOUS REPRESENTATIVE ALGORITHMS WITH BOCHNER KERNELS AND OUR RFF-DIGMM MODEL WITH THE POLYNOMIAL KERNEL. THE BEST SCORES ARE HIGHLIGHTED BY **BOLDFACE**

Dataset	RF	Recursive-Nyström	CROiclassification	RFF-DIGMM
	learned kernel	Gaussian kernel	Gaussian kernel	polynomial kernel
	Acc:% (time:sec.)	Acc:% (time:sec.)	Acc:% (time:sec.)	Acc:% (time:sec.)
ijcnn1	95.5±0.4 (36.6±2.3)	94.5±0.0 (6.2±0.2)	97.1±0.1 (0.4±0.1)	**97.8**±0.1 (10.7±0.9)
covtype	79.2±0.28 (44.4±2.2)	81.2±0.8 (480.3±25.5)	**86.2**±0.2 (17.7±0.9)	83.1±0.3 (19.8±2.2)
skin	97.9±0.2 (40.4±1.4)	**98.5**±0.8 (5.8±0.2)	98.1±0.1 (3.4±0.3)	98.2±0.1 (3.1±1.3)
EEG	84.3±0.8 (5.0±0.1)	84.8±0.5 (0.3±0.2)	**87.2**±0.1 (0.4±0.1)	85.1±0.5 (4.0±0.8)
spambase	86.2±1.4 (2.0±0.2)	84.9±8.3 (0.2±0.1)	90.2±0.1 (0.2±0.1)	**90.9**±0.5 (6.2±0.2)

inference. Here, we quantitatively study the influence of the size of the sketch, i.e., $N_s = 1, 5, 10, 50, 100$ in our method with the polynomial kernel and TL1 kernel on the ijcnn1 data set.

Fig. 3 shows the kernel approximation error, test accuracy, and time cost for the polynomial kernel approximation varying with different sizes of the sketch. We can see that if more data points are sampled, our method with the polynomial kernel

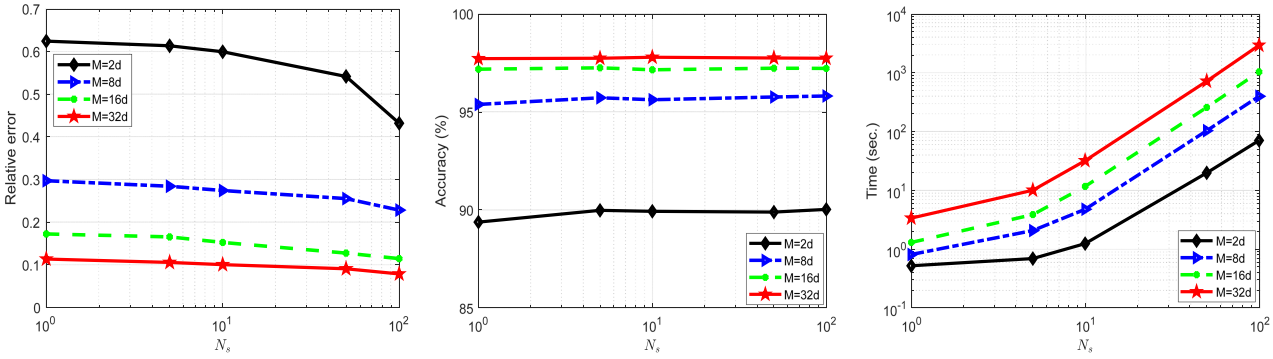


Fig. 3. Comparison of (a) approximation error, (b) test accuracy, and (c) time cost for kernel approximation of varying N_s for the polynomial kernel on the ijcnn1 data set. (a) Relative error. (b) Test accuracy. (c) Time cost.

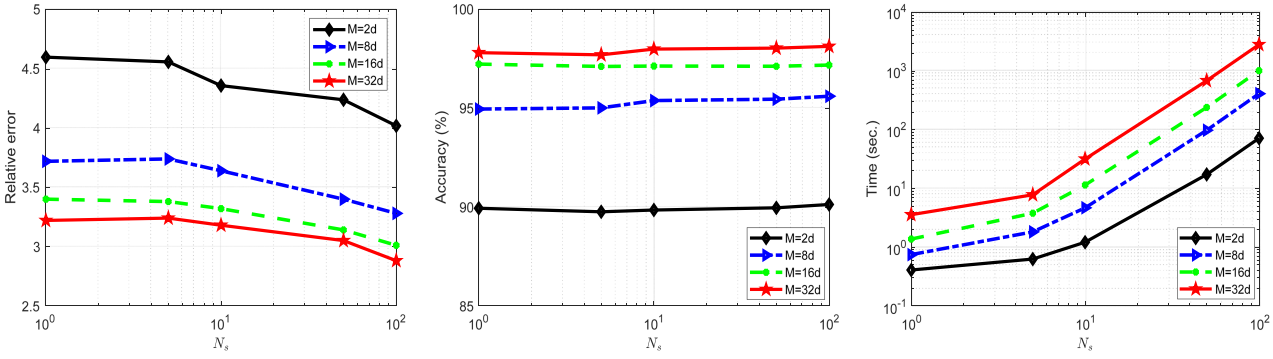


Fig. 4. Comparison of (a) approximation error, (b) test accuracy, and (c) time cost for kernel approximation of varying N_s for the TL1 kernel on the ijcnn1 data set. (a) Relative error. (b) Test accuracy. (c) Time cost.

TABLE VII
COMPARISON RESULTS OF DIFFERENT TRUNCATION PARAMETER VALUES ON THE ijcnn1 DATA SET

kernel type	T	$M = 2d$	$M = 4d$	$M = 8d$	$M = 16d$	$M = 32d$
		Acc:% (time:sec.)	Acc:% (time:sec.)	Acc:% (time:sec.)	Acc:% (time:sec.)	Acc:% (time:sec.)
Polynomial kernel	$T = 1$	89.9±0.6 (0.2±0.2)	92.8±0.9 (0.4±0.2)	95.7±0.4 (0.6±0.2)	97.0±0.4 (1.0±0.2)	98.1±0.3 (2.4±0.2)
	$T = 5$	90.0±0.6 (0.7±0.4)	92.1±0.7 (1.1±0.6)	95.7±0.4 (2.1±0.8)	97.3±0.4 (3.9±0.9)	97.8±0.1 (10.7±0.9)
	$T = 10$	89.7±0.2 (0.8±0.4)	91.9±0.8 (1.1±0.7)	94.8±0.7 (2.7±0.9)	97.4±0.4 (5.3±0.7)	98.2±0.2 (12.0±0.5)
TL1 kernel	$T = 1$	90.1±0.7 (0.2±0.2)	92.2±0.8 (0.3±0.2)	95.3±0.5 (0.5±0.2)	97.0±0.2 (0.8±0.4)	97.8±0.3 (1.3±0.6)
	$T = 5$	89.8±0.5 (0.6±0.4)	91.7±0.6 (1.0±0.5)	95.0±0.4 (1.8±0.9)	97.1±0.3 (3.8±0.8)	97.4±0.3 (7.8±0.7)
	$T = 10$	89.9±0.3 (0.7±0.4)	92.0±0.7 (1.1±0.7)	94.9±0.5 (2.0±0.9)	96.9±0.3 (4.1±0.8)	97.8±0.2 (11.8±0.7)

achieves slight improvements on the kernel approximation error and the test accuracy. However, in terms of computational cost, the training time significantly increases along with more sampled data taken into consideration, as shown in Fig. 3. In addition, Fig. 4 shows that our model with the TL1 kernel achieves the same tendencies with the polynomial kernel setting, in terms of the approximation error, classification accuracy, and time cost.

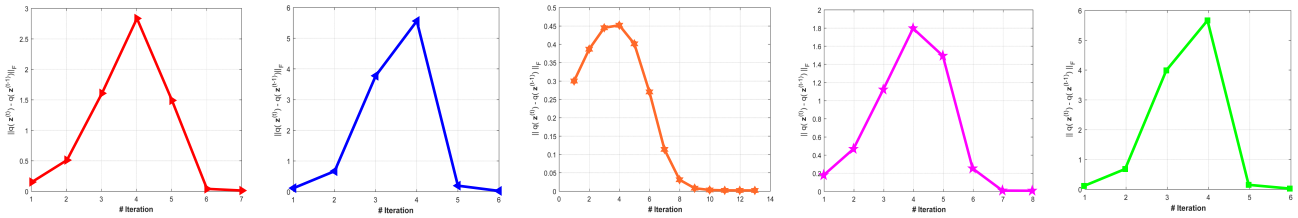
From the above-mentioned experimental results, although the sketch with larger size would lead to better approximation performance to some extent, this strategy cannot guarantee better classification performance. This might be because the original kernel might not be suitable for the task, as discussed in [3] and [60].

2) Truncation Parameter: As aforementioned, the variational distribution is truncated, but our model is a full DP and is not truncated. The truncation level T is a variational parameter that can be freely set; it is not a part of the prior model specification. Here, we evaluate the parametric sensitivity of T on the ijcnn1 data set. Table VII reports the classification accuracy and the time cost for computing random features when T is chosen as 1, 5, and 10. It can be observed that the test accuracy with a different T is experimentally stable. However, the time cost gradually increases as T rises. Hence, small T values are shown to achieve high computational efficiency, which explains the reason why we choose this parameter for our experiments.

3) Eigenvalue Decomposition Parameter: Here, we investigate how the eigenvalue decomposition parameter τ in (5) does affect the final performance. We evaluate our model with the TL1 kernel ($M = 2d$) on the ijcnn1 data set under a different τ. In our experiment, τ is chosen as $-1.1\mu_N$, $-2\mu_N$,

TABLE VIII
COMPARISON RESULTS WITH DIFFERENT EIGENVALUE DECOMPOSITION PARAMETER τ ON THE ijcnn1 DATA SET

τ	$-1.1\mu_N$	$-2\mu_N$	$-5\mu_N$	$-10\mu_N$	$-20\mu_N$
Acc:%(time:sec.)	89.8±0.5 (0.6±0.4)	89.6±0.6 (0.7±0.3)	89.0±0.4 (0.7±0.5)	89.5±0.5 (0.6±0.3)	89.9±0.4 (0.7±0.5)



(a) (b) (c) (d) (e)

Fig. 5. Convergence plots on (a) ijcnn1, (b) covtype, (c) *skin*, (d) EEG, and (e) spambase.

$-5\mu_N$, $-10\mu_N$, and $-20\mu_N$, where $\mu_N < 0$ is the smallest eigenvalue of the indefinite kernel matrix $\boldsymbol{K}$. Table VIII shows that different selections of τ do not have a significant influence on the final classification accuracy and the time cost for kernel approximation. Hence, the performance is insensitive to the parameter τ, and thus, we experimentally set it to $-1.1\mu_N$ in our model.

F. Illustration of Convergence

Here, we investigate the convergence of the used non-conjugate variational inference algorithm. We take the TL1 kernel with $M = 2d$ as an example and plot $\|q(\boldsymbol{z}^t) - q(\boldsymbol{z}^{(t-1)})\|_{\mathrm{F}}$ versus iteration on the above-mentioned five data sets in Fig. 5. It can be found that, in most cases, $q(\boldsymbol{z}^t)$ significantly decays in the first five iterations in our variational inference algorithm, which leads to a quick convergence under the stopping criterion $\|q(\boldsymbol{z}^t) - q(\boldsymbol{z}^{(t-1)})\|_{\mathrm{F}} \leq 1e^{-5}$. The total iterations are less than ten in these five data sets except for *skin* with about 13 iterations. Therefore, the maximum iteration number fixed to 50 is reasonable and enough. Furthermore, the convergence of the optimization process employed by our non-conjugate variational inference is well demonstrated.

VI. CONCLUSION

We investigated a full non-parametric Bayesian method in random feature mappings for indefinite kernels. It extends the traditional Bochner kernel in RFF to several non-Bochner kernels, including dot-product kernels and indefinite kernels. By placing a DP prior to the components of Gaussian mixtures, our RFF-DIGMM model is adaptive to the data with varying components. The derived non-conjugate variational inference algorithm with the sub-sampling scheme is efficient and effective for model inference. As a result, the superiority of our method is demonstrated by experimental validation on several classification data sets.

APPENDIX
UPDATE VARIATIONAL FACTORS

The optimization for each variational factor is conducted by iteratively updating the latent variables in detail.

1) $q(\beta_t)$: We absorb terms in (13) that do not depend on β_t into the additive normalization constant, giving

$$\begin{aligned}\ln q^*(\beta_t) &= \mathbb{E}_{\Omega\setminus\beta_t} \ln p(\mathcal{D}_s, \Omega) + \text{const} \\ &= \ln p(\beta_t) + \sum_{m=1}^{M} \mathbb{E}_q[\ln p(z_m|\tilde{\boldsymbol{\beta}})] + c\,.\end{aligned}$$

Following [56] and $q(z_m > T) = 0$, we have:

$$\begin{aligned}\mathbb{E}_q[\ln p(z_m|\tilde{\boldsymbol{\beta}})] = \sum_{k=1}^{T}\{&q(z_m > k)\mathbb{E}_q[\ln(1-\beta_k)] \\ &+ q(z_m = k)\mathbb{E}(\ln \beta_k)\}\,.\end{aligned}$$

As a result, the optimal variational distribution $q^*(\beta_t)$ can be obtained by

$$\begin{aligned}\ln q^*(\beta_t) &= \ln p(\beta_t) + \sum_{m=1}^{M}[q(z_m > t)\ln(1-\beta_t) \\ &\qquad + q(z_m = t)\ln\beta_t] + c \\ &= \ln p(\beta_t) + \left[\sum_{m=1}^{M} q(z_m > t)\right]\ln(1-\beta_t) \\ &\quad + \left[\sum_{m=1}^{M} q(z_m = t)\right]\ln\beta_t + c.\end{aligned}$$

Since $\beta_k \sim \text{Beta}(1, \alpha_0)$, we have $p(\beta_k) \propto (1-\beta_k)^{\alpha_0 - 1}$. Finally, we have

$$\beta_t \sim \text{Beta}\left(1 + \sum_{m=1}^{M} q(z_m = t), \alpha_0 + \sum_{m=1}^{M} q(z_m > t)\right).$$

2) $q(z_m)$: Likewise, we do not consider irrelevant terms of z_m in (13), that is

$$\begin{aligned}&\ln q^*(z_m) \\ &= \mathbb{E}_{\Omega\setminus z_m} \ln p(\mathcal{D}_s, \Omega) + c \\ &= \mathbb{E}_q[\ln p(z_m|\tilde{\boldsymbol{\beta}}) + \ln p(\boldsymbol{w}_m|z_m, \tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\Lambda}})] + c \\ &= \sum_{k=1}^{T}\Bigg\{1[z_m > k]\mathbb{E}[\ln(1-\beta_k)] + 1[z_m = k]\mathbb{E}(\ln\beta_k) \\ &\qquad + 1(z_m = k)\left(\frac{1}{2}\mathbb{E}[\ln|\boldsymbol{\Lambda}_k|] - \frac{d}{2}\ln(2\pi) - \frac{1}{2}\Xi\right)\Bigg\} + c\end{aligned}$$

with $\Xi \triangleq \mathbb{E}_{\boldsymbol{w}_m,\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k}[(\boldsymbol{w}_m-\boldsymbol{\mu}_k)^\top \boldsymbol{\Lambda}_k(\boldsymbol{w}_m-\boldsymbol{\mu}_k)]$. Defining

$$\ln \hbar_{mk} = \mathbb{E}(\ln \beta_k) + \sum_{t=1}^{k-1}\mathbb{E}[\ln(1-\beta_t)] + \frac{1}{2}(\mathbb{E}\ln|\boldsymbol{\Lambda}_k| - d\ln(2\pi) - \Xi)$$

and scaling $\tilde{\hbar}_{mk} = (\hbar_{mk}/\sum_{t=1}^{T}\hbar_{mt})$, we have $q(z_m = k) = \tilde{\hbar}_{mk}$. It means that z_m is chosen according to a multinomial probability distribution.

3) $q(\boldsymbol{\mu}_k)$*:* Keeping only terms that have a functional dependence on $\boldsymbol{\mu}_k$, we have

$$\begin{aligned}
&\ln q^*(\boldsymbol{\mu}_k)\\
&= \mathbb{E}_{\Omega\setminus\boldsymbol{\mu}_k}\ln p(\mathcal{D}_s,\boldsymbol{\Omega}) + c\\
&= \mathbb{E}_q\left[\ln p(\boldsymbol{\mu}_k) + \ln\prod_{m=1}^{M} p(\boldsymbol{w}_m|z_m,\tilde{\boldsymbol{\mu}},\tilde{\boldsymbol{\Lambda}})\right] + c\\
&= \ln p(\boldsymbol{\mu}_k) + \sum_{m=1}^{M} q(z_m=k)\mathbb{E}_{\Omega\setminus\boldsymbol{\mu}_k}\left[\ln\mathcal{N}\left(\boldsymbol{w}_m|\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k^{-1}\right)\right] + c\\
&= -\frac{1}{2}\boldsymbol{\mu}_k^\top\left(\boldsymbol{R}_0 + \mathbb{E}(\boldsymbol{\Lambda}_k)\sum_{m=1}^{M} q(z_m=k)\right)\boldsymbol{\mu}_k\\
&\quad + \boldsymbol{\mu}_k^\top\left(\boldsymbol{R}_0\boldsymbol{m}_0 + \mathbb{E}(\boldsymbol{\Lambda}_k)\sum_{m=1}^{M} q(z_m=k)\mathbb{E}(\boldsymbol{w}_m)\right).
\end{aligned}$$

After some algebraic manipulations, as we expect, $\boldsymbol{\mu}_k$ is subject to a Gaussian distribution $\boldsymbol{\mu}_k \sim \mathcal{N}(\boldsymbol{\mu}_k|\boldsymbol{m}_k,\boldsymbol{R}_k^{-1})$ with the following mean vector and precision matrix:

$$\begin{cases}
\boldsymbol{m}_k = \boldsymbol{R}_k^{-1}\left(\boldsymbol{R}_0\boldsymbol{m}_0 + \mathbb{E}(\boldsymbol{\Lambda}_k)\sum_{m=1}^{M} q(z_m=k)\mathbb{E}(\boldsymbol{w}_m)\right)\\
\boldsymbol{R}_k = \boldsymbol{R}_0 + \mathbb{E}(\boldsymbol{\Lambda}_k)\sum_{m=1}^{M} q(z_m=k).
\end{cases}$$

4) $q(\boldsymbol{\Lambda}_k)$*:* We only retain some terms with respect to $\boldsymbol{\Lambda}_k$ in (13), namely

$$\begin{aligned}
&\ln q^*(\boldsymbol{\Lambda}_k)\\
&= \mathbb{E}_{\Omega\setminus\boldsymbol{\Lambda}_k}\ln p(\mathcal{D}_s,\boldsymbol{\Omega}) + c\\
&= \ln p(\boldsymbol{\Lambda}_k) + \sum_{m=1}^{M} q(z_m=k)\mathbb{E}_{\Omega\setminus\boldsymbol{\Lambda}_k}\left[\ln\mathcal{N}\left(\boldsymbol{w}_m|\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k^{-1}\right)\right] + c\\
&= -\frac{1}{2}\mathrm{Tr}\left(\boldsymbol{\Lambda}_k\boldsymbol{W}_0^{-1}\right) + \frac{\nu_0-d-1}{2}\ln|\boldsymbol{\Lambda}_k|\\
&\quad + \frac{1}{2}\left(\sum_{m=1}^{M} q(z_m=k)\right)\ln|\boldsymbol{\Lambda}_k|\\
&\quad - \frac{1}{2}\sum_{m=1}^{M} q(z_m=k)\mathrm{Tr}\left(\boldsymbol{\Lambda}_k\mathbb{E}(\boldsymbol{w}_m-\boldsymbol{\mu}_k)(\boldsymbol{w}_m-\boldsymbol{\mu}_k)^\top\right) + c.
\end{aligned}$$

Thus, we have $\boldsymbol{\Lambda}_k \sim \mathcal{W}(\boldsymbol{\Lambda}_k|\boldsymbol{W}_k,\nu_k)$ with $\nu_k = \nu_0 + \sum_{m=1}^{M} q(z_m=k)$, and $\boldsymbol{W}_k^{-1}$ is formulated by

$$\boldsymbol{W}_k^{-1} = \boldsymbol{W}_0^{-1} + \sum_{m=1}^{M} q(z_m=k)\mathbb{E}(\boldsymbol{w}_m-\boldsymbol{\mu}_k)(\boldsymbol{w}_m-\boldsymbol{\mu}_k)^\top.$$

5) $q(\boldsymbol{w}_m)$*:* Inspecting (13) and reading off those terms that involve only $\boldsymbol{w}_m$, we have

$$\begin{aligned}
\ln q^*(\boldsymbol{w}_m) &= \mathbb{E}_{\Omega\setminus\boldsymbol{w}_m}\ln p(\mathcal{D}_s,\boldsymbol{\Omega}) + c\\
&= \mathbb{E}_q[\ln p(\boldsymbol{w}_m|z_m,\tilde{\boldsymbol{\mu}},\tilde{\boldsymbol{\Lambda}})]\\
&\quad + \mathbb{E}_q\left[\ln\prod_{i,j=1,i\neq j}^{N_s} p\left(K_{ij}^+|(\boldsymbol{x}_i,\boldsymbol{x}_j),\boldsymbol{w}_m\right)\right] + c.
\end{aligned}$$

For the first term, we have

$$\begin{aligned}
&\mathbb{E}_q[\ln p(\boldsymbol{w}_m|z_m,\tilde{\boldsymbol{\mu}},\tilde{\boldsymbol{\Lambda}})]\\
&= \mathbb{E}_q[\ln p(\boldsymbol{w}_m|\tilde{\boldsymbol{\mu}},\tilde{\boldsymbol{\Lambda}})^{1[z_m=k]}]\\
&= \frac{1}{2}\sum_{k=1}^{T} q(z_m=k)(\mathbb{E}[\ln|\boldsymbol{\Lambda}_k|]\\
&\qquad - \mathbb{E}_{\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k}[(\boldsymbol{w}_m-\boldsymbol{\mu}_k)^\top\boldsymbol{\Lambda}_k(\boldsymbol{w}_m-\boldsymbol{\mu}_k)]) + c.
\end{aligned}$$

The second term can be expressed as

$$\begin{aligned}
&\mathbb{E}_q\left[\ln\prod_{i,j=1,i\neq j}^{N_s} p\left(K_{ij}^+|(\boldsymbol{x}_i,\boldsymbol{x}_j),\boldsymbol{w}_m\right)\right]\\
&\qquad = -\frac{1}{2\sigma_\epsilon^2}\sum_{i,j=1,i\neq j}^{N_s}\left(K_{ij}^+ - \cos\left[\boldsymbol{w}_m^\top(\boldsymbol{x}_i-\boldsymbol{x}_j)\right]\right)^2 + c.
\end{aligned}$$

Since $\boldsymbol{w}_m$ is not a conjugate variable, we conduct the second-order Taylor expansion $\cos[\boldsymbol{w}_m^\top(\boldsymbol{x}_i-\boldsymbol{x}_j)] \approx 1 - (1/2)\boldsymbol{w}_m^\top(\boldsymbol{x}_i-\boldsymbol{x}_j)(\boldsymbol{x}_i-\boldsymbol{x}_j)^\top\boldsymbol{w}_m$ and derive that

$$\ln q^*(\boldsymbol{w}_m) \approx -\frac{1}{2}\boldsymbol{w}_m^\top\boldsymbol{S}\boldsymbol{w}_m + \boldsymbol{w}_m^\top\sum_{k=1}^{T}[q(z_m=k)\mathbb{E}(\boldsymbol{\Lambda}_k)\mathbb{E}(\boldsymbol{\mu}_k)] + c.$$

where $\boldsymbol{S}$ is defined by

$$\boldsymbol{S} = \sum_{k=1}^{T}[q(z_m=k)\mathbb{E}(\boldsymbol{\Lambda}_k)] + \frac{1}{2\sigma_\epsilon^2}\sum_{i,j=1,i\neq j}^{N_s}\left(1-K_{ij}^+\right)(\boldsymbol{x}_i-\boldsymbol{x}_j)(\boldsymbol{x}_i-\boldsymbol{x}_j)^\top.$$

Therefore, $\boldsymbol{w}_m$ is subject to

$$\boldsymbol{w}_m \sim \mathcal{N}\left(\boldsymbol{S}^{-1}\left\{\sum_{k=1}^{T}[q(z_m=k)\mathbb{E}(\boldsymbol{\Lambda}_k)\mathbb{E}(\boldsymbol{\mu}_k)]\right\},\boldsymbol{S}^{-1}\right).$$

In the variational update equations, we also need to calculate the expectations with respect to the current variational distributions. For example, $\mathbb{E}(\boldsymbol{\mu}_k)$ and $\mathbb{E}(\boldsymbol{\Lambda}_k)$ can be easily obtained by their respective distributions. Here, we present several intractable expectation computations. The expectation $\mathbb{E}(\ln|\boldsymbol{\Lambda}_k|)$ can be obtained by

$$\mathbb{E}(\ln|\boldsymbol{\Lambda}_k|) = \sum_{i=1}^{d}\psi\left(\frac{\nu_k+1-i}{2}\right) + d\ln 2 + \ln|\boldsymbol{W}_k|$$

where $\psi(\cdot)$ is the digamma function with $\psi(x) = (\mathrm{d}/\mathrm{d}x)\ln\Gamma(x)$. Besides, the expectation

$\mathbb{E}_{\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k}[(\boldsymbol{w}_m-\boldsymbol{\mu}_k)^\top\boldsymbol{\Lambda}_k(\boldsymbol{w}_m-\boldsymbol{\mu}_k)]$ is given by

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k}\big[(\boldsymbol{w}_m-\boldsymbol{\mu}_k)^\top\boldsymbol{\Lambda}_k(\boldsymbol{w}_m-\boldsymbol{\mu}_k)\big]\\
&=\int_{\boldsymbol{\mu}_k}\int_{\boldsymbol{\Lambda}_k}\mathrm{Tr}\big[\boldsymbol{\Lambda}_k(\boldsymbol{w}_m-\boldsymbol{\mu}_k)^\top(\boldsymbol{w}_m-\boldsymbol{\mu}_k)\big]q^*(\boldsymbol{\mu}_k)q^*(\boldsymbol{\Lambda}_k)\mathrm{d}\boldsymbol{\mu}_k\mathrm{d}\boldsymbol{\Lambda}_k\\
&=\int_{\boldsymbol{\Lambda}_k}\int_{\boldsymbol{\mu}_k}\mathrm{Tr}\big[\boldsymbol{\Lambda}_k\big(\boldsymbol{w}_m^\top\boldsymbol{w}_m-2\boldsymbol{w}_m^\top\boldsymbol{\mu}_k+\boldsymbol{\mu}_k^\top\boldsymbol{\mu}_k\big)\big]q^*(\boldsymbol{\mu}_k)\mathrm{d}\boldsymbol{\mu}_k q^*(\boldsymbol{\Lambda}_k)\mathrm{d}\boldsymbol{\Lambda}_k\\
&=\int_{\boldsymbol{\Lambda}_k}\mathrm{Tr}\big[\boldsymbol{\Lambda}_k\big(\boldsymbol{w}_m^\top\boldsymbol{w}_m-2\boldsymbol{w}_m^\top\boldsymbol{m}_k+\boldsymbol{m}_k^2+\boldsymbol{R}_k^{-1}\big)\big]q^*(\boldsymbol{\Lambda}_k)\mathrm{d}\boldsymbol{\Lambda}_k\\
&=\mathbb{E}(\boldsymbol{\Lambda}_k)\big[(\boldsymbol{w}_m-\boldsymbol{m}_k)^\top(\boldsymbol{w}_m-\boldsymbol{m}_k)+\boldsymbol{R}_k^{-1}\big].
\end{aligned}$$

Similarly, the expectations $\mathbb{E}_{\boldsymbol{w}_m,\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k}[(\boldsymbol{w}_m-\boldsymbol{\mu}_k)^\top\boldsymbol{\Lambda}_k(\boldsymbol{w}_m-\boldsymbol{\mu}_k)]$ and $\mathbb{E}_{\boldsymbol{w}_m,\boldsymbol{\mu}_k}[(\boldsymbol{w}_m-\boldsymbol{\mu}_k)^\top\boldsymbol{\Lambda}_k(\boldsymbol{w}_m-\boldsymbol{\mu}_k)]$ can be calculated by the above-mentioned way.

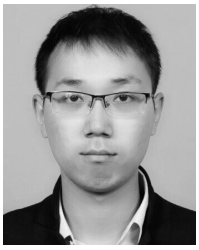
ACKNOWLEDGMENT

The authors sincerely appreciate the anonymous reviewers for their insightful comments.

REFERENCES

[1] P. Kar and H. Karnick, "Random feature maps for dot product kernels," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2012, pp. 583–591.
[2] N. Pham and R. Pagh, "Fast and scalable polynomial kernels via explicit feature maps," in *Proc. ACM Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 239–247.
[3] J. Pennington, X. Y. Felix, and S. Kumar, "Spherical random features for polynomial kernels," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1846–1854.
[4] A. Sinha and J. C. Duchi, "Learning kernels with random features," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 1298–1306.
[5] C. Musco and C. Musco, "Recursive sampling for the Nyström method," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3833–3845.
[6] M. Kafai and K. Eshghi, "CROification: Accurate kernel classification with the efficiency of sparse linear SVM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 1, pp. 34–48, Jan. 2019.
[7] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.
[8] J. Sethuraman, "A constructive definition of Dirichlet priors," *Statistica Sinica*, vol. 4, no. 2, pp. 639–650, 1994.
[9] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, "Stochastic variational inference," *J. Mach. Learn. Res.*, vol. 14, pp. 1303–1347, May 2013.
[10] S. I. Amari, "Natural gradient works efficiently in learning," *Neural Comput.*, vol. 10, no. 2, pp. 251–276, 1998.
[11] C. Wang and D. M. Blei, "Variational inference in nonconjugate models," *J. Mach. Learn. Res.*, vol. 14, pp. 1005–1031, Jan. 2013.
[12] C.-J. Hsieh, S. Si, and I. Dhillon, "A divide-and-conquer solver for kernel support vector machines," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 566–574.
[13] Y. Zhang, J. Duchi, and M. Wainwright, "Divide and conquer kernel ridge regression," in *Proc. Conf. Learn. Theory*, 2013, pp. 592–617.
[14] A. J. Smola and B. Schölkopf, "Sparse greedy matrix approximation for machine learning," in *Proc. Int. Conf. Mach. Learn.*, 2000, pp. 911–918.
[15] S. Fine and K. Scheinberg, "Efficient SVM training using low-rank kernel representations," *J. Mach. Learn. Res.*, vol. 2, no. 2, pp. 243–264, 2001.
[16] C. K. Williams and M. Seeger, "Using the Nyström method to speed up kernel machines," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 682–688.
[17] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 1177–1184.
[18] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. Evanston, IL, USA: Routledge, 2018.
[19] B. Schölkopf and A. J. Smola, *Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2003.
[20] A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing, "Deep kernel learning," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2016, pp. 370–378.
[21] C. M. Alaíz, M. Fanuel, and J. A. K. Suykens, "Convex formulation for kernel PCA and its use in semisupervised learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3863–3869, Jun. 2018.
[22] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer, 1995.
[23] H. Drucker, C. J. Burges, L. Kaufman, A. J. Smola, and V. Vapnik, "Support vector regression machines," in *Proc. Adv. Neural Inf. Process. Syst.*, 1997, pp. 155–161.
[24] B. Schölkopf, A. Smola, and K.-R. Müller, "Kernel principal component analysis," in *Proc. Int. Conf. Artif. Neural Netw.*, 1997, pp. 583–588.
[25] Y. Sun, A. Gilbert, and A. Tewari, "But how does it work in theory? Linear SVM with random features," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 3383–3392.
[26] S. Mehrkanoon and J. A. K. Suykens, "Deep hybrid neural-kernel networks using random Fourier features," *Neurocomputing*, vol. 298, pp. 46–54, Jul. 2018.
[27] K. Cutajar, E. V. Bonilla, P. Michiardi, and M. Filippone, "Random feature expansions for deep Gaussian Gaussian processes," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 884–893.
[28] B. Xie, Y. Liang, and L. Song, "Scale up nonlinear component analysis with doubly stochastic gradients," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2341–2349.
[29] D. Lopez-Paz, S. Sra, A. Smola, Z. Ghahramani, and B. Schölkopf, "Randomized nonlinear component analysis," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1359–1367.
[30] S. Bochner, *Harmonic Analysis and the Theory of Probability*. Chelmsford, MA, USA: Courier Corporation, 2005.
[31] C. S. Ong, X. Mary, and A. J. Smola, "Learning with non-positive kernels," in *Proc. Int. Conf. Mach. Learn.*, 2004, pp. 81–89.
[32] F. Liu, X. Huang, C. Gong, J. Yang, and J. A. K. Suykens, "Indefinite kernel logistic regression with concave-inexact-convex procedure," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 3, pp. 765–776, Mar. 2019.
[33] G. Loosli, S. Canu, and S. O. Cheng, "Learning SVM in Kreĭn spaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 6, pp. 1204–1216, Jun. 2016.
[34] A. Vedaldi and A. Zisserman, "Efficient additive kernels via explicit feature maps," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 3, pp. 480–492, Mar. 2012.
[35] A. J. Smola, Z. L. Ovari, and R. C. Williamson, "Regularization with dot-product kernels," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 308–314.
[36] X. Huang, J. A. K. Suykens, S. Wang, J. Hornegger, and A. Maier, "Classification with truncated ℓ_1 distance kernel," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 5, pp. 2025–2030, May 2018.
[37] A. Feragen, F. Lauze, and S. Hauberg, "Geodesic exponential kernels: When curvature and linearity conflict," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3032–3042.
[38] R. Hamid, Y. Xiao, A. Gittens, and D. Decoste, "Compact random feature maps," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 19–27.
[39] S. Roweis and Z. Ghahramani, "A unifying review of linear Gaussian models," *Neural Comput.*, vol. 11, no. 2, pp. 305–345, 1999.
[40] J. Bognár, *Indefinite Inner Product Spaces*. Berlin, Germany: Springer, 1974.
[41] F. X. Yu, A. T. Suresh, K. Choromanski, D. Holtmannrice, and S. Kumar, "Orthogonal random features," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 1975–1983.
[42] M. I. Jordan and M. J. Wainwright, "Graphical models, exponential families, and variational inference," *Found. Trends Mach. Learn.*, vol. 1, nos. 1–2, pp. 1–305, 2007.
[43] S. J. Gershman and D. M. Blei, "A tutorial on Bayesian nonparametric models," *J. Math. Psychol.*, vol. 56, no. 1, pp. 1–12, 2012.
[44] H. Avron, V. Sindhwani, J. Yang, and M. W. Mahoney, "Quasi-Monte Carlo feature maps for shift-invariant kernels," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 4096–4133, 2016.
[45] Q. Le, T. Sarlós, and A. Smola, "FastFood—Approximating kernel expansions in loglinear time," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1–9.
[46] M. Munkhoeva, Y. Kapushev, E. Burnaev, and I. Oseledets, "Quadrature-based features for kernel approximation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 9165–9174.
[47] T. S. Ferguson, "A Bayesian analysis of some nonparametric problems," *Ann. Statist.*, vol. 1, no. 2, pp. 209–230, 1973.
[48] C. E. Rasmussen, "The infinite Gaussian mixture model," in *Proc. Adv. Neural Inf. Process. Syst.*, 2000, pp. 554–560.

[49] P. Orbanz and Y. W. Teh, *Bayesian Nonparametric Models*. New York, NY, USA: Springer, 2011.
[50] I. Steinwart and C. Andreas, *Support Vector Machines*. Springer, 2008.
[51] Z. C. Guo and L. Shi, "Optimal rates for coefficient-based regularized regression," *Appl. Comput. Harmon. Anal.*, to be published. doi: 10.1016/j.acha.2017.11.005.
[52] J. B. Oliva, A. Dubey, A. G. Wilson, B. Pöczos, J. Schneider, and E. P. Xing, "Bayesian nonparametric kernel learning," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2016, pp. 1078–1086.
[53] V. Maz'ya and G. Schmidt, "On approximate approximations using Gaussian kernels," *IMA J. Numer. Anal.*, vol. 16, no. 1, pp. 13–29, 1996.
[54] M. Luby and A. Wigderson, "Pairwise independence and derandomization," *Found. Trends Theor. Comput. Sci.*, vol. 1, no. 4, pp. 237–301, 2006.
[55] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *J. Amer. Stat. Assoc.*, vol. 112, no. 518, pp. 859–877, 2016.
[56] D. M. Blei and M. I. Jordan, "Variational inference for Dirichlet process mixtures," *Bayesian Anal.*, vol. 1, no. 1, pp. 121–143, 2006.
[57] C. Blake and C. J. Merz. (1998). *UCI Repository of Machine Learning Databases*. [Online]. Available: http://archive.ics.uci.edu/ml/
[58] 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, Nov. 1998.
[59] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, Aug. 2008.
[60] F. X. Yu, S. Kumar, H. Rowley, and S. F. Chang, "Compact nonlinear maps and circulant extensions," 2015. *arXiv:1503.03893*. [Online]. Available: https://arxiv.org/abs/1503.03893



Fanghui Liu (M'19) received the B.E. degree in automation from the Harbin Institute of Technology, Harbin, China, in 2014, and the Ph.D. degree from the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai, China, in 2019.

He is currently a Post-Doctoral Researcher with ESAT-STADIUS, KU Leuven, Leuven, Belgium. His current research interests include kernel methods, learning theory, and optimization and applications to computer vision.

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, and the Ph.D. degree in control science and engineering from Tsinghua University, Beijing, China, in 2012.

From 2012 to 2015, he was a Post-Doctoral Researcher with ESAT-STADIUS, KU Leuven, Leuven, Belgium. After that, he was selected as an Alexander von Humboldt Fellow and working at the Pattern Recognition Laboratory, the Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany, where he was appointed as the Group Head. Since 2016, he has been an Associate Professor with the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai, China. His current research interests include machine learning, optimization, and their applications.

Dr. Huang has been awarded as "1000-Talent" (Young Program) in 2017.

Lei Shi received the joint Ph.D. degree in applied mathematics from the City University of Hong Kong, Hong Kong, and the University of Science and Technology of China, Hefei, China, in 2010.

He is currently an Associate Professor with the School of Mathematical Sciences, Fudan University, Shanghai, China. His current research interests include learning theory and approximation theory.

Jie Yang received the Ph.D. from the Department of Computer Science, Hamburg University, Hamburg, Germany, in 1994.

He is currently a Professor with the Institute of Image Processing and Pattern recognition, Shanghai Jiao Tong University, Shanghai, China. He has led many research projects (e.g., the National Science Foundation, 863 National High Tech. Plan), had one book published in Germany, and authored more than 300 journal papers. His current research interests include object detection and recognition, data fusion and data mining, and medical image processing.

Johan A. K. Suykens (SM'05–F'15) was born in Willebroek, Belgium, in May 18, 1966. He received the M.S. degree in electro-mechanical engineering and the Ph.D. degree in applied sciences from Katholieke Universiteit Leuven, Leuven, Belgium, in 1989 and 1995, respectively.

In 1996, he was a Visiting Post-Doctoral Researcher with the University of California at Berkeley, Berkeley, CA, USA. He has been a Post-Doctoral Researcher with the Fund for Scientific Research FWO Flanders, Belgium. He is currently a Professor (Hoogleraar) with KU Leuven, Leuven. He is currently serving as the Program Director of Master AI at KU Leuven. He is the 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), a coauthor of the book, *Cellular Neural Networks, Multi-Scroll Chaos and Synchronization* (World Scientific), and an Editor of the books, *Nonlinear Modeling: Advanced Black-Box Techniques* (Kluwer Academic Publishers) and *Advances in Learning Theory: Methods, Models and Applications* (IOS Press).

Dr. Suykens has been an elevated IEEE Fellow 2015 for developing least squares support vector machines. He has been awarded an ERC Advanced Grant 2011 and 2017. He was a recipient of the International Neural Networks Society INNS 2000 Young Investigator Award for significant contributions in the field of neural networks. He received the IEEE Signal Processing Society 1999 Best Paper (Senior) Award and several best paper awards at international conferences. In 1998, he organized an International Workshop on Nonlinear Modeling with Time-Series Prediction Competition. He has served as Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS from 1997 to 1999 and 2004 to 2007 and the IEEE TRANSACTIONS ON NEURAL NETWORKS from 1998 to 2009. He has served as the Director and an Organizer of the NATO Advanced Study Institute on Learning Theory and Practice (Leuven, in 2002), a Program Co-Chair for the 2004 International Joint Conference on Neural Networks and the 2005 International Symposium on Nonlinear Theory and its Applications, an Organizer of the 2007 International Symposium on Synchronization in Complex Networks, a Co-Organizer of the NIPS 2010 workshop on Tensors, Kernels and Machine Learning, and the chair of the International Workshop on Advances in Regularization, Optimization, Kernel Methods, and Support Vector Machines: Theory and Applications (ROKS) 2013.