Deep Spectral Kernel Learning

Hui Xue[1,2] , **Zheng-Fan Wu**[1,2] and **Wei-Xiang Sun**[1,2]

[1]School of Computer Science and Engineering, Southeast University, Nanjing, 210096, China
[2]MOE Key Laboratory of Computer Network and Information Integration (Southeast University), China
{hxue, zfwu, vex-soon}@seu.edu.cn

Abstract

Recently, spectral kernels have attracted wide attention in complex dynamic environments. These advanced kernels mainly focus on breaking through the crucial limitation on *locality*, that is, the stationarity and the monotonicity. But actually, owing to the inefficiency of shallow models in computational elements, they are more likely unable to accurately reveal dynamic and potential variations. In this paper, we propose a novel deep spectral kernel network (DSKN) to naturally integrate non-stationary and non-monotonic spectral kernels into *elegant* deep architectures in an interpretable way, which can be further generalized to cover most kernels. Concretely, we firstly deal with the general form of spectral kernels by the inverse Fourier transform. Secondly, DSKN is constructed by embedding the preeminent spectral kernels into each layer to boost the efficiency in computational elements, which can effectively reveal the dynamic input-dependent characteristics and potential long-range correlations by compactly representing complex advanced concepts. Thirdly, detailed analyses of DSKN are presented. Owing to its universality, we propose a unified spectral transform technique to flexibly extend and reasonably initialize domain-related DSKN. Furthermore, the representer theorem of DSKN is given. Systematical experiments demonstrate the superiority of DSKN compared to state-of-the-art relevant algorithms on varieties of standard real-world tasks.

1 Introduction

Kernel method is a class of artful statistical learning approaches. Benefiting from their flexible modeling frameworks and excellent statistical theories, they have been successfully used in many traditional learning applications over the past few decades. However, with the rapid development of machine learning in recent years, most classic kernels are no longer applicable to complex tasks in practical dynamic environments. In fact, many theoretical and experimental analyses have shown that the most critical and fundamental limitation of these kernels is that they are *local*, that is, stationary and monotonic [Bengio *et al.*, 2006]. For instance, the most widely-used *local* Gaussian kernel $k(\boldsymbol{x}, \boldsymbol{x}')$ only considers the distance $\|\boldsymbol{x} - \boldsymbol{x}'\|$ and quickly converges to a constant when $\|\boldsymbol{x} - \boldsymbol{x}'\|$ increases. Consequently, it can't reveal more essential information over feature spaces except for the identical similarity [Remes *et al.*, 2017].

In order to solve such a problem, some new kinds of kernels have been presented to break the restriction on *locality*. They generally fall into three categories [Rasmussen and Williams, 2006]: (1) non-stationary kernels; (2) non-monotonic kernels; (3) non-stationary and non-monotonic kernels. In the first category, these kernels only focus on non-stationarity, which are constructed by mapping input spaces [Sampson and Guttorp, 1992] or taking input-dependent parameters [Gibbs, 1998; Heinonen *et al.*, 2016]. However, they can't adequately reflect pivotal long-range data correlations due to neglecting the non-monotonicity. On the contrary, the kernels in the second category only focus on the non-monotonicity, which are generated by solving the inverse Fourier transform based on Bochner's theorem [Wilson and Adams, 2013; Lázaro-Gredilla *et al.*, 2010]. But they lose essential input-dependent characteristics that can be learned well by non-stationary kernels.

Just recently, several improved spectral kernels are proposed to acquire the non-stationarity and the non-monotonicity simultaneously following the generalized Fourier analysis theory about kernels [Yaglom, 1987]. Remes *et al.* [2017] derived a generalized spectral mixture kernel by defining the spectral density as bivariate Gaussian mixture components. Ton *et al.* [2018] directly solved the inverse Fourier transform using Monte Carlo approximation, and proposed a generalized sparse spectrum kernel. To some extent, these innovative spectral kernels make up for the deficiency caused by *locality*. But, due to the inefficiency of shallow models in computational elements, they still can't effectively and efficiently acquire the appropriate non-stationary and non-monotonic properties when learning problems become more and more complicated. That likely leads to their poor performance in real-world tasks.

Although Bochner's theorem [Gikhman and Skorohod, 1974] and Yaglom's theorem [Yaglom, 1987] theoretically guarantee that spectral kernels can availably approximate most kernels under specific conditions, they actually require exponential computational elements to represent complex

kernels [Rahimi and Recht, 2008], which restricts the further development and application of spectral kernels. Indeed, this serious problem is ubiquitous in most shallow models, not only spectral kernels, but also shallow neural networks [Delalleau and Bengio, 2011]. In neural networks, deep architectures have been introduced to obtain the exponential improvement in computational elements from hierarchical nonlinear linking structures [Bengio *et al.*, 2007]. Consequently, in order to break the limitation on *locality* and improve the efficiency in computational elements, it is quite necessary to apply deep architectures to spectral kernels.

In this paper, we focus on effectively and flexibly integrating the non-stationary and non-monotonic spectral kernels into well-designed deep architectures in a natural and interpretable manner. Specifically, we firstly deal with the spectral kernel based on Yaglom's theorem and derive a more general spectral form. DSKN is then formulated and constructed by embedding the distinguished spectral kernels into each layer to boost the efficiency in computational elements, which benefits a lot from the *elegant* deep architecture, including universality, flexibility, efficiency and interpretability. It can be further generalized to cover most kernels naturally. Detailed analyses of DSKN are presented later. We propose a unified spectral transform technique to flexibly extend and reasonably initialize DSKN due to the universality of the deep architecture, and thus we can dynamically inject some vital priori information to construct domain-related DSKN. The representer theorem and the reproducing kernel Hilbert space of DSKN are also derived recursively. More importantly, DSKN strengthens the link between kernel method and deep learning. It can not only improve spectral kernels by compactly representing highly non-linear and highly-varying advanced concepts, but also significantly enhance the performance of deep learning in medium-scale and small-scale tasks, especially when the deep architectures can't be well constructed from the complex but sparse data distribution. Systematical experiments demonstrate the superiority of DSKN compared to state-of-the-art relevant algorithms on varieties of standard classification and regression tasks, which indicates that the proposed approach adequately learns from the intrinsic advantages of spectral kernels and deep architectures.

2 Related Work

Recently, some deep kernel algorithms have been presented to try to link kernel method with deep learning. Most of them directly combine frequently-used deep modules as the front-end or back-end of kernels. Wilson *et al.* [2016b] placed a plain deep neural network as the front-end of a spectral mixture kernel to extract features, which is further extended to a structured kernel interpolation framework [Wilson and Nickisch, 2015] and stochastic variational inference [Wilson *et al.*, 2016a]. Sun *et al.* [2018] used a sum-product network as the back-end of multiple kernels to merge the kernel mappings. However, the deep modules and the kernels are relatively detached in these algorithms and thus they can't intrinsically improve the efficiency of kernels in computational elements.

Some other algorithms stack kernel mappings in a hierarchical composite way. Cho and Saul [2009] designed a class of arc-cosine kernels and integrated them into a deep hierarchical structure. Zhuang *et al.* [2011] proposed the 2-layer multiple kernel structure which further leads to a series of refined algorithms [Rebai *et al.*, 2016]. But these above models are relatively closed and inflexible, which are only suitable for the specific kernels and difficult to be optimized.

Moreover, there are some algorithms that aim to introduce kernels into deep learning and further construct end-to-end complete deep models. Stacked kernel network is derived by replacing the non-linear activation functions of deep neural network with kernel mappings [Zhang *et al.*, 2017]. Deep Gaussian process combines multiple Gaussian processes hierarchically [Cutajar *et al.*, 2017]. But, the models derived from these methods can't be regarded as kernel functions anymore, and thus can't be simply applied to kernel algorithms.

3 Deep Spectral Kernel Learning

In this section, we firstly introduce some brief concepts about the spectral kernels with the non-stationary and non-monotonic properties. Subsequently, we explicitly formulate and construct the novel DSKN by hierarchically stacking the kernel mappings of the derived spectral kernels in a scalable manner. Furthermore, we present detailed analyses of DSKN and propose a unified spectral transform technique to flexibly extend and reasonably initialize domain-related DSKN. The representer theorem and the reproducing kernel Hilbert space of DSKN are also derived recursively.

3.1 Spectral Kernel

Spectral kernels are constructed from the inverse Fourier transform in frequency domain. Most commonly-used spectral kernels are stationary, such as spectral mixture kernel [Wilson and Adams, 2013] and sparse spectrum kernel [Lázaro-Gredilla *et al.*, 2010]. These stationary kernels are shift-invariant functions that only depend on the distance $\boldsymbol{\tau} = \boldsymbol{x} - \boldsymbol{x}'$ of inputs $\boldsymbol{x}$ and $\boldsymbol{x}'$, and thus can be rewritten as $k(\boldsymbol{x}, \boldsymbol{x}') = k(\boldsymbol{x} - \boldsymbol{x}') = k(\boldsymbol{\tau})$. A stationary kernel $k(\boldsymbol{\tau})$ can be uniquely identified by a spectral density $s(\boldsymbol{\omega})$ on the basis of Bochner's theorem [Gikhman and Skorohod, 1974; Stein, 1999]:

$$\begin{aligned} k(\boldsymbol{\tau}) &= \int_{\mathbb{R}^D} e^{i\boldsymbol{\omega}^T\boldsymbol{\tau}} s(\boldsymbol{\omega}) d\boldsymbol{\omega}, \\ s(\boldsymbol{\omega}) &= \int_{\mathbb{R}^D} e^{-i\boldsymbol{\omega}^T\boldsymbol{\tau}} k(\boldsymbol{\tau}) d\boldsymbol{\tau}, \end{aligned} \tag{1}$$

where $s(\boldsymbol{\omega})$ is the spectral density of a non-negative measure. Some important stationary kernels and corresponding spectral densities are shown in Table 1 [Rahimi and Recht, 2008]. Spectral kernels $k(\boldsymbol{\tau}) = k(\boldsymbol{x} - \boldsymbol{x}')$ derived from the theorem are non-monotonic but stationary, which can learn potential long-range relationships but neglect the important dynamic input-dependent characteristics.

Recently, the Fourier analysis theory of kernels have achieved tremendous improvements. Yaglom's theorem further indicates that a general kernel $k(\boldsymbol{x}, \boldsymbol{x}')$ is related to a spectral density $s(\boldsymbol{\omega}, \boldsymbol{\omega}')$ in accordance with the following

Kernel	$k(\boldsymbol{\tau})$	$s(\boldsymbol{\omega})$
Gaussian	$e^{-\frac{\|\boldsymbol{\tau}\|_2^2}{2}}$	$(2\pi)^{-\frac{D}{2}} e^{-\frac{\|\boldsymbol{\omega}\|_2^2}{2}}$
Laplacian	$e^{-\|\boldsymbol{\tau}\|_1}$	$\prod_{i=1}^{D} \frac{1}{\pi(1+\boldsymbol{\omega}_i^2)}$
Cauchy	$\prod_{i=1}^{D} \frac{2}{1+\boldsymbol{\tau}_i^2}$	$e^{-\|\boldsymbol{\tau}\|_1}$

Table 1: Important stationary kernels and spectral densities

Fourier duals:

$$\begin{aligned} k(\boldsymbol{x},\boldsymbol{x}') &= \int_{\mathbb{R}^D\times\mathbb{R}^D} e^{i(\boldsymbol{\omega}^T\boldsymbol{x}-\boldsymbol{\omega}'^T\boldsymbol{x}')} s(\boldsymbol{\omega},\boldsymbol{\omega}')d\boldsymbol{\omega}d\boldsymbol{\omega}', \\ s(\boldsymbol{\omega},\boldsymbol{\omega}') &= \int_{\mathbb{R}^D\times\mathbb{R}^D} e^{-i(\boldsymbol{\omega}^T\boldsymbol{x}-\boldsymbol{\omega}'^T\boldsymbol{x}')} k(\boldsymbol{x},\boldsymbol{x}')d\boldsymbol{x}d\boldsymbol{x}', \end{aligned} \tag{2}$$

where the kernel $k(\boldsymbol{x},\boldsymbol{x}')$ is positive semi-definite if and only if $s(\boldsymbol{\omega},\boldsymbol{\omega}')$ is the positive semi-definite bounded variation spectral density of a Lebesgue-Stieltjes measure [Yaglom, 1987]. As a result, spectral kernels constructed by Eq. (2) can not only learn long-range relationships but also important input-dependent characteristics, benefiting from their *non-locality*, that is, the non-stationarity and the non-monotonicity.

3.2 Deep Spectral Kernel Network

Many theoretical and experimental researches have indicated that deep models have significant efficiency superiority than shallow counterparts in computational elements [Delalleau and Bengio, 2011]. Inspired by the intrinsic superiority of deep learning, we explicitly design and construct DSKN through two progressive steps. We firstly deal with the non-stationary and non-monotonic spectral kernels, and derive a more general spectral form. Then we naturally and flexibly integrate the solved spectral kernels into *elegant* deep architectures in a hierarchical composite way.

Following Eq. (2), we symmetrize the integral to alleviate the restriction of Yaglom's theorem. Concretely, the exponential component $e^{i(\boldsymbol{\omega}^T\boldsymbol{x}-\boldsymbol{\omega}'^T\boldsymbol{x}')}$ in Eq. (2) is replaced by an augmented part $\mathcal{E}_{\boldsymbol{\omega},\boldsymbol{\omega}'}(\boldsymbol{x},\boldsymbol{x}')$:

$$\begin{aligned} \mathcal{E}_{\boldsymbol{\omega},\boldsymbol{\omega}'}(\boldsymbol{x},\boldsymbol{x}') = \frac{1}{4}\Big[&e^{i(\boldsymbol{\omega}^T\boldsymbol{x}-\boldsymbol{\omega}'^T\boldsymbol{x}')} + e^{i(\boldsymbol{\omega}'^T\boldsymbol{x}-\boldsymbol{\omega}^T\boldsymbol{x}')} \\ &+e^{i(\boldsymbol{\omega}^T\boldsymbol{x}-\boldsymbol{\omega}^T\boldsymbol{x}')} + e^{i(\boldsymbol{\omega}'^T\boldsymbol{x}-\boldsymbol{\omega}'^T\boldsymbol{x}')}\Big]. \end{aligned} \tag{3}$$

As a result, the spectral surface density $s(\boldsymbol{\omega},\boldsymbol{\omega}')$ and the corresponding spectral surface $S(\boldsymbol{\omega},\boldsymbol{\omega}')$ can be further regarded as a continuous probability density and a corresponding cumulative distribution function, respectively [Ton *et al.*, 2018; Rahimi and Recht, 2008]. In other words, we can directly optimize $\boldsymbol{\omega},\boldsymbol{\omega}'$ over the $\mathbb{R}^D\times\mathbb{R}^D$ Euclidean space.

Subsequently, the inverse Fourier transform of $s(\boldsymbol{\omega},\boldsymbol{\omega}')$ is solved as follows:

$$\begin{aligned} k(\boldsymbol{x},\boldsymbol{x}') &= \int_{\mathbb{R}^D\times\mathbb{R}^D} \mathcal{E}_{\boldsymbol{\omega},\boldsymbol{\omega}'}(\boldsymbol{x},\boldsymbol{x}')s(\boldsymbol{\omega},\boldsymbol{\omega}')d\boldsymbol{\omega}d\boldsymbol{\omega}' \\ &= \mathbb{E}_{\boldsymbol{\omega},\boldsymbol{\omega}'\sim S}\big[\mathcal{E}_{\boldsymbol{\omega},\boldsymbol{\omega}'}(\boldsymbol{x},\boldsymbol{x}')\big] \\ &= \mathbb{E}_{\boldsymbol{\omega},\boldsymbol{\omega}'\sim S}\big[\mathcal{T}_{\boldsymbol{\omega},\boldsymbol{\omega}'}(\boldsymbol{x},\boldsymbol{x}')\big], \end{aligned} \tag{4}$$

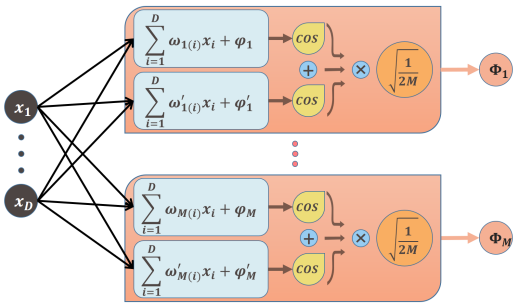

Figure 1: The structure of the kernel mapping $\Phi(\boldsymbol{x})$

where $\mathcal{T}_{\boldsymbol{\omega},\boldsymbol{\omega}'}(\boldsymbol{x},\boldsymbol{x}')$ is:

$$\begin{aligned} \frac{1}{4}\Big[&\cos(\boldsymbol{\omega}^T\boldsymbol{x}-\boldsymbol{\omega}'^T\boldsymbol{x}') + \cos(\boldsymbol{\omega}'^T\boldsymbol{x}-\boldsymbol{\omega}^T\boldsymbol{x}') \\ &+\cos(\boldsymbol{\omega}^T\boldsymbol{x}-\boldsymbol{\omega}^T\boldsymbol{x}') + \cos(\boldsymbol{\omega}'^T\boldsymbol{x}-\boldsymbol{\omega}'^T\boldsymbol{x}')\Big]. \end{aligned} \tag{5}$$

Consequently, we derive the non-stationary and non-monotonic spectral kernel $k(\boldsymbol{x},\boldsymbol{x}')$ by directly approximating the expectation with Monte Carlo integral:

$$k(\boldsymbol{x},\boldsymbol{x}') = \mathbb{E}_{\boldsymbol{\omega},\boldsymbol{\omega}'\sim S}\big[\mathcal{T}_{\boldsymbol{\omega},\boldsymbol{\omega}'}(\boldsymbol{x},\boldsymbol{x}')\big] \approx \langle\Psi(\boldsymbol{x}),\Psi(\boldsymbol{x}')\rangle, \tag{6}$$

where the spectral kernel mapping Ψ is:

$$\Psi(\boldsymbol{x}) = \sqrt{\frac{1}{4M}}\begin{bmatrix}\cos(\boldsymbol{\Omega}^T\boldsymbol{x})+\cos(\boldsymbol{\Omega}'^T\boldsymbol{x}) \\ \sin(\boldsymbol{\Omega}^T\boldsymbol{x})+\sin(\boldsymbol{\Omega}'^T\boldsymbol{x})\end{bmatrix}. \tag{7}$$

M is the sampling number. The $D\times M$ frequency matrices $\boldsymbol{\Omega},\boldsymbol{\Omega}'$ are denoted as:

$$\boldsymbol{\Omega} = \big[\boldsymbol{\omega}_1,\cdots,\boldsymbol{\omega}_M\big], \boldsymbol{\Omega}' = \big[\boldsymbol{\omega}'_1,\cdots,\boldsymbol{\omega}'_M\big]. \tag{8}$$

The frequency pairs $\{(\boldsymbol{\omega}_i,\boldsymbol{\omega}'_i)\}_{i=1}^{M} \overset{i.i.d.}{\sim} S$ [Ton *et al.*, 2018].

In DSKN, we further adopt another more general form Φ instead of Ψ:

$$\Phi(\boldsymbol{x}) = \sqrt{\frac{1}{2M}}\Big[\cos(\boldsymbol{\Omega}^T\boldsymbol{x}+\boldsymbol{\varphi})+\cos(\boldsymbol{\Omega}'^T\boldsymbol{x}+\boldsymbol{\varphi}')\Big], \tag{9}$$

which is equivalent to Ψ in the sense of expectation that $\mathbb{E}[\Phi(\boldsymbol{x})] = \mathbb{E}[\Psi(\boldsymbol{x})]$. The phase vectors $\boldsymbol{\varphi}$ and $\boldsymbol{\varphi}'$ are drawn uniformly from $[0,2\pi]^M$. Compared with Ψ, the adopted form Φ can halve the computational overhead and alleviate the difficulty in programming. The detailed structure of the spectral kernel mapping $\Phi(\boldsymbol{x})$ is illustrated in Figure 1. Without losing generality, it can be regarded as a slightly more complex neural network with only single hidden layer and using cosine as the activation function.

Therefore, according to Eq. (9), the non-stationary and non-monotonic spectral kernel $k(\boldsymbol{x},\boldsymbol{x}')$ can be identified by a pair of frequency matrices $\boldsymbol{\Omega},\boldsymbol{\Omega}'$ sampled from the spectral surface S. It's worthy to point out that we can not only approximate almost all kernels by assigning specific $\boldsymbol{\Omega},\boldsymbol{\Omega}'$, but also derive more powerful spectral kernels by dynamically optimizing $\boldsymbol{\Omega},\boldsymbol{\Omega}'$.

Although some fundamental theorems guarantee that spectral kernels shown in Eq. (9) and Figure 1 can availably approximate most kernels under specific conditions [Gikhman

and Skorohod, 1974; Yaglom, 1987], their learning capability for representing complex kernels in practical dynamic environments requires exponential computational elements $\Phi_1, \cdots, \Phi_M$ [Rahimi and Recht, 2008]. Fortunately, considering the superiority of deep learning in improving the efficiency of computational elements, we further effectively and flexibly integrate spectral kernels $k(\boldsymbol{x}, \boldsymbol{x}')$ into well-designed deep architectures in an interpretable way.

Actually, one of the most serious problems in existing deep kernels is that they can't pertinently construct flexible deep structures for kernels. The reason more likely lies in the form restrictions of kernels. That is, a kernel function is a bivariate scalar function and its kernel mapping is implicit. However, benefiting from the general spectral form of the spectral kernels shown in Eq. (9) and Figure 1, we directly avoid the above limitations to naturally construct DSKN by stacking Φ in a deep hierarchical composite way:

$$\mathcal{K}^{(l)}(\boldsymbol{x}, \boldsymbol{x}') = \langle \Phi^l(\Phi^{l-1}(\cdots \Phi^1(\boldsymbol{x}))), \Phi^l(\Phi^{l-1}(\cdots \Phi^1(\boldsymbol{x}'))) \rangle, \tag{10}$$

where $\mathcal{K}^{(l)}$ denotes the l-layer DSKN, and Φ^l denotes the l-th layer spectral kernel mapping. The parentheses (l) here mean that the l-th layer and the previous layers are both included. We further simplify the notation of $\mathcal{K}^{(l)}$ as:

$$\mathcal{K}^{(l)}(\boldsymbol{x}, \boldsymbol{x}') = \langle \Xi^{(l)}(\boldsymbol{x}), \Xi^{(l)}(\boldsymbol{x}') \rangle, \tag{11}$$

where the composite kernel mapping $\Xi^{(l)}$ of DSKN is:

$$\Xi^{(l)}(\boldsymbol{x}) = \Phi^l(\Phi^{l-1}(\cdots \Phi^1(\boldsymbol{x}))). \tag{12}$$

Based on the structure of Φ illustrated in Figure 1, the complete architecture of the l-layer DSKN $\mathcal{K}^{(l)}$ is shown as Figure 2.

3.3 Analysis of DSKN

The deep architecture of DSKN illustrated in Figure 2 is significantly more *natural* and *elegant* than those closed structures of existing deep kernels. DSKN benefits a great deal from the concise hierarchical architecture, including universality, flexibility, efficiency and interpretability.

Specifically, DSKN is an unprecedented universal deep kernel framework that can be directly applied to most positive semi-definite kernels, without losing any generality, to construct their composite deep models. In other words, we can assign all kinds of positive semi-definite kernels as the internal elements of DSKN with the unified spectral transform technique. Concretely, given a kernel $\hat{k}$ to be embedded, we firstly derive its spectral density $\hat{s}$ and spectral surface $\hat{S}$ by solving the inverse Fourier transform in Eq. (1) or Eq. (2), where $\hat{k}$ is uniquely identified by $\hat{s}$ and $\hat{S}$. Then, based on the frequency pairs $\{(\boldsymbol{\omega}_i, \boldsymbol{\omega}_i')\}_{i=1}^M \overset{i.i.d.}{\sim} \hat{S}$ and Eq. (9), we further construct a spectral kernel $\tilde{k}$ to approximate the kernel $\hat{k}$. Consequently, we can flexibly embed any positive semi-definite kernel $\hat{k}$ by indirectly integrating spectral kernel $\tilde{k} \approx \hat{k}$. What's more, these essentially different kernels $\hat{k}$ can be uniformly optimized. Some commonly-used kernels and corresponding spectral densities have been shown in Table 1. By doing so, according to practical applications, we can naturally assign well-designed kernels as the internal elements of DSKN by the unified spectral transform technique, and thus more pivotal priori knowledge can be reasonably injected to construct domain-related DSKN.

Furthermore, DSKN can be flexibly adjusted in vertical and horizontal manners. That is to say, for the vertical adjustment, we can increase the depth l of DSKN by directly integrating more spectral kernels into the network. For the horizontal adjustment, in addition to directly increasing the sampling number M, actually, each layer Φ^i for $i = 1, \cdots, l$ can be further replaced by an augmented heterogeneous multiple kernel structure including K components $\{\phi_k^i\}_{k=1}^K$:

$$\Phi^i(\cdot) = \left[\phi_1^i(\cdot)^T, \phi_2^i(\cdot)^T, \cdots, \phi_K^i(\cdot)^T\right]^T, \tag{13}$$

where each element $\phi_k^i(\cdot)$ is essentially a complete spectral kernel mapping shown in Eq. (9) and Figure 1. Therefore, DSKN can effectively embed different kernels into any position and easily adjust the deep network. Considering that all parameters in DSKN are represented as frequencies ω, DSKN can be uniformly optimized by most existing optimization algorithms in deep learning, such as SGD and Adam, in accordance with the error backpropagation. Furthermore, in Θ notation, the computational complexity of DSKN is the same as that of classic deep neural networks with the same architectures.

Compared with other deep kernels, the architecture of DSKN is more interpretable by explicitly stacking spectral kernel mappings in a hierarchical composite way. We derive the representer theorem and the reproducing kernel Hilbert space $\mathcal{V}^{(i)}$ of the i-layer DSKN $\mathcal{K}^{(i)}$ for all $i = 1, \cdots, l$ recursively [Bohn *et al.*, 2017]. Concretely, let $\mathcal{H}^i$ be the reproducing kernel Hilbert space of the mapping Φ^i about the kernel k^i with finite-dimensional domain D^i and range R^i where $R^i \subseteq \mathbb{R}^{M_i}$ with $M_i \in \mathbb{N}$ for $i = 1, \cdots, l$. Such that $R^{i-1} \subseteq D^i$ for $i = 2, \cdots, l$ and $D^1 \subseteq \mathbb{R}^D$. Let $\mathcal{L}$ be an arbitrary loss function and $\Theta^1, \cdots, \Theta^l$ be strictly monotonically increasing functions. The minimization objective $\mathcal{J}$ is defined as:

$$\begin{aligned} \mathcal{J}(\Phi^1, \cdots, \Phi^l) = &\sum_{n=1}^N \mathcal{L}(\Phi^l(\Phi^{l-1}(\cdots \Phi^1(\boldsymbol{x}_n))), y_n) \\ &+ \sum_{i=1}^l \Theta^i(\|\Phi^i\|_{\mathcal{H}^i}^2). \end{aligned} \tag{14}$$

Then, a set of minimizers $\{\Phi^i\}_{i=1}^l$ with $\Phi^i \subseteq \mathcal{H}^i$ fulfills $\Xi^{(i)}(\cdot) = \Phi^i(\Phi^{i-1}(\cdots \Phi^1(\cdot))) \in \mathcal{V}^{(i)} \subset \mathcal{H}^i$ for all $i = 1, \cdots, l$ with the spanned reproducing kernel Hilbert space $\mathcal{V}^{(i)}$ of the i-layer DSKN $\mathcal{K}^{(i)}$:

$$\begin{aligned} \mathcal{V}^{(i)} = \operatorname{span}\big\{ &k^i(\Phi^{i-1}(\Phi^{i-2}(\cdots \Phi^1(\boldsymbol{x}_n))), \cdot)\boldsymbol{e}_{m_i} \\ &| n = 1, \cdots N; m_i = 1, \cdots M_i \big\}, \end{aligned} \tag{15}$$

where $\boldsymbol{e}_{m_i} \in \mathbb{R}^{M_i}$ is the m_i-th unit vector. Intuitively, according to the image spaces of hidden layers, the kernels in previous layers try to align the intrinsic features of data in such a way that they can be easily resolved by the posterior

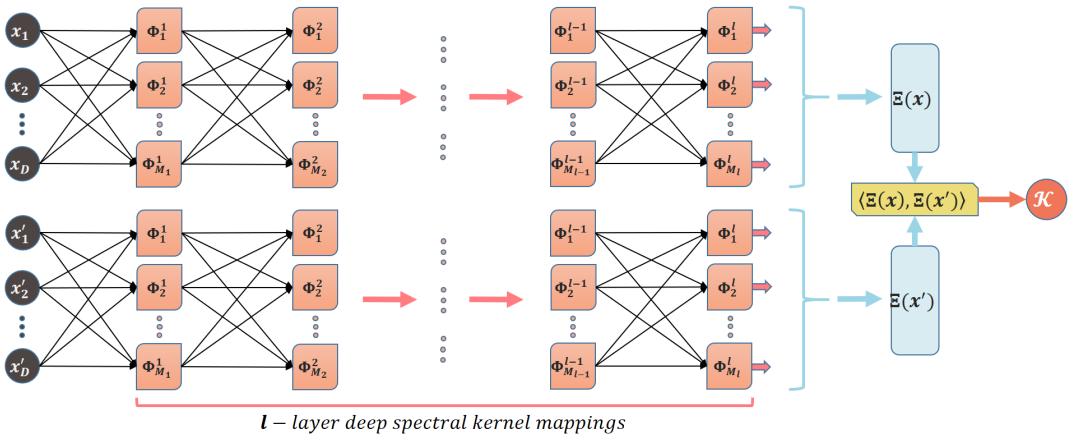

Figure 2: The deep architecture of DSKN

ones, and thus the output of the last layer in DSKN can represent more important advanced concepts for tasks at hand.

It is worth noting that DSKN further strengthens the link between kernel method and deep learning. On the one hand, it is distinctly superior to shallow spectral kernels in all aspects by effectively applying deep architectures to spectral kernels, which significantly improve the learning capability of spectral kernels to compactly represent highly non-linear and highly-varying complex concepts by abstracting the advanced internal relationships of inputs. On the other hand, compared with deep learning, it can adapt to medium-scale and small-scale learning tasks better, by naturally embedding the powerful non-stationary and non-monotonic spectral kernels into deep networks, which can incisively learn dynamic input-dependent characteristics and potential long-range correlations across the whole feature spaces.

4 Experiments

In this section, we experimentally evaluate the performance of DSKN compared with several state-of-the-art algorithms on varieties of typical tasks, which demonstrates that DSKN can achieve all-round performance improvements.

4.1 Experimental Setup

As the priori knowledge, the most widely-used classic Gaussian kernels are used as the internal basic kernel elements of DSKN, whose spectral surfaces S are Gaussian distributions. Moreover, the scales of all deep architectures in the experiments are uniformly set to $1000 \times 500 \times 50$. Sigmoid is applied to the activation functions in neural networks. DSKN and the compared kernels are applied to the same Gaussian process models for classification and regression, and optimized by Adam. The accuracy and the mean absolute error (MAE) are chosen as the evaluation criteria for classification tasks and regression tasks, respectively, which reflect the average performance better.

Compared Algorithms

DSKN is compared with several state-of-the-art relevant algorithms including:

- **1-SKN** [Ton *et al.*, 2018]: 1-layer Spectral Kernel Network is a shallow baseline, and the number of computational elements is set to be the same as DSKN.
- **DNN** [LeCun *et al.*, 1998]: Deep Neural Network is the most classic model in deep learning.
- **DKL-GA** [Wilson *et al.*, 2016b]: Deep Kernel Learning with GAussian kernel combines a DNN as the front-end of a Gaussian kernel.
- **DKL-SM** [Wilson *et al.*, 2016b]: Deep Kernel Learning with Spectral Mixture kernel combines a DNN as the front-end of a spectral mixture kernel.

Datasets

We systematically evaluate the performance of DSKN on several standard classification and regression tasks. We firstly conduct classification experiments on four benchmark datasets, including *four*, *ionosphere*, *splice* and *wbdc* [Blake and Merz, 1998]. Secondly, we conduct regression experiments on other four datasets including *airfoil*, *boston*, *concrete* and *energy* [Blake and Merz, 1998]. All these data are scaled by z-score standardization and randomly divided into two non-overlapping training and test sets, which are equal in size. The division, training and test processes are repeated ten times to generate ten independent results for each dataset, and then we assess the average performance with corresponding evaluation criterion. Furthermore, to evaluate the performance of DSKN on training data with different scales, we specifically conduct an image classification experiment on *MNIST* dataset [LeCun *et al.*, 1998]. The division of training data and test data is consistent with the default scheme. Specifically, 10,000 images are randomly selected as test data, and the rest are training data. The training data are further sampled to different scales from 5% to 100%.

4.2 Experimental Results

We evaluate the performance of DSKN in varieties of standard classification and regression tasks. Experimental results are shown in Table 2, where the best results are highlighted in bold. ($\uparrow$) indicates the larger the better, while ($\downarrow$) indicates

	Classification Accuracy (↑)				Regression MAE (↓)			
	four	*ionosphere*	*splice*	*wbdc*	*airfoil*	*boston*	*concrete*	*energy*
1-SKN	0.984±0.021●	0.864±0.065●	0.674±0.064●	0.964±0.023	0.282±0.010●	0.248±0.019	0.269±0.010●	0.121±0.013●
DNN	0.864±0.143●	0.828±0.102●	0.777±0.082●	0.932±0.106	0.198±0.009●	0.290±0.018●	0.237±0.014	0.088±0.008●
DKL-GA	0.896±0.157●	0.743±0.115●	0.682±0.134●	0.902±0.147	0.246±0.033●	0.249±0.016	0.239±0.012	0.099±0.006●
DKL-SM	0.963±0.106	0.787±0.105●	0.748±0.082●	0.940±0.103	0.280±0.020●	0.257±0.024	0.258±0.018●	0.104±0.004●
DSKN	**0.999±0.000**	**0.917±0.033**	**0.855±0.012**	**0.974±0.006**	**0.176±0.012**	**0.244±0.016**	**0.233±0.012**	**0.069±0.003**

Table 2: Classification accuracy and regression MAE (mean±std) of each compared algorithm on several real-world datasets. (↑) indicates the larger the better, while (↓) indicates the smaller the better. The best results are highlighted in bold. In addition, ●/○ indicates whether DSKN is statistically superior/inferior to the compared algorithms on each dataset (pairwise t-test at 0.05 significance level).

	1-SKN	**DNN**	**DKL-GA**	**DKL-SM**	**DSKN**
5%	0.9052	0.9589	0.9498	0.9437	**0.9762**
10%	0.9312	0.9733	0.9714	0.9595	**0.9824**
20%	0.9479	0.9817	0.9820	0.9782	**0.9881**
40%	0.9614	0.9862	0.9858	0.9866	**0.9920**
70%	0.9726	0.9876	0.9874	0.9865	**0.9936**
100%	0.9746	0.9891	0.9899	0.9881	**0.9945**

Table 3: Accuracy on *MNIST* dataset with different scales.

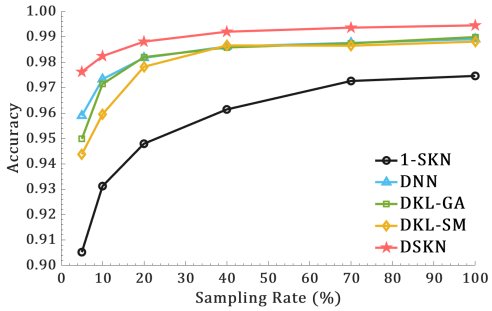

Figure 3: Accuracy curves on *MNIST* dataset with different scales.

the smaller the better. In order to measure the significance of performance difference statistically, pairwise t-test at 0.05 significance level is conducted. Specifically, when DSKN is significantly superior/inferior to the compared algorithms, a marker ●/○ is denoted [Xu *et al.*, 2017].

According to the results illustrated in Table 2, 1-SKN with only single hidden layer performs relatively well on most easy classification tasks benefiting from the non-stationary and non-monotonic properties to some extent. But the shallow 1-SKN lacks the efficiency in computational elements, and thus performs poorly on regression tasks which need to be learned more accurately. As a classic and commonly-used deep learning algorithms, DNN is still very competitive on average. The two compared deep kernels, DKL-GA and DKL-SM, have relatively poor performance. Although additional kernels are combined as the back-end, these traditional deep kernels can't achieve effective performance improvements. By contrast, the proposed DSKN evidently outperforms all compared algorithms on all tasks. The experimental results explicitly demonstrate the excellent performance and stability of DSKN, which further implicitly indicates the necessity of effectively construct *natural* deep architectures for non-stationary and non-monotonic spectral kernels.

We further conduct an experiment on *MNIST* dataset with different scales to demonstrate the excellent learning ability of DSKN. The results are collected in Table 3 and visualized as Figure 3. On the relatively complex dataset, 1-SKN performs poorly due to its shallow structure, which needs exponential computational elements to availably represent image patterns. There is no significant performance gap between DNN, DKL-GA and DKL-SM. But the deep kernels, DKL-GA and DKL-SM, still perform a little bit worse than DNN. The kernels in the last layer are more likely to affect the adequate backpropagation of error information. By contrast, DSKN can not only stably achieve the best performance on all scales, but also enlarge the performance superiority over compared algorithms on medium-scale and small-scale tasks. In fact, the sparser the data distribution, the more crucial the appropriate non-stationary and non-monotonic properties are. DSKN accurately learns the crucial properties and thus performs better, by introducing deep architectures to improve the non-stationary and non-monotonic spectral kernels.

5 Conclusion

In view of the prominent superiority of deep architectures over shallow ones, we pay attention to effectively integrate non-stationary and non-monotonic spectral kernels into *elegant* deep architectures, and propose the novel DSKN, which can be further generalized to cover most kernels. Specifically, we firstly deal with the spectral kernels by inverse Fourier transform and present a more general spectral form. DSKN is then derived by naturally embedding the spectral kernels into each layer to achieve better efficiency in computational elements. Consequently, DSKN can effectively reveal the dynamic input-dependent characteristics and potential long-range correlations by compactly representing complex advanced concepts. In addition, some intuitive analyses and the representer theorem of DSKN are also presented. Systematical experiments demonstrate the superiority of DSKN compared to state-of-the-art relevant algorithms on varieties of standard tasks, which implicitly indicates that DSKN can relatively adopt more strong points of spectral kernels and deep architectures while overcoming their weak points.

Acknowledgments

This work was supported by the National Key R&D Program of China (2017YFB1002801), the National Natural Science Foundations of China (Grant No. 61876091). It is also supported by Collaborative Innovation Center of Wireless Communications Technology.

References

[Bengio *et al.*, 2006] Yoshua Bengio, Olivier Delalleau, and Nicolas Le Roux. The curse of highly variable functions for local kernel machines. In *Advances in neural information processing systems*, pages 107–114, 2006.

[Bengio *et al.*, 2007] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. In *Advances in neural information processing systems*, pages 153–160, 2007.

[Blake and Merz, 1998] Catherine Blake and Christopher J Merz. Uci repository of machine learning databases. *Online at http://archive.ics.uci.edu/ml/*, 1998.

[Bohn *et al.*, 2017] Bastian Bohn, Michael Griebel, and Christian Rieger. A representer theorem for deep kernel learning. *arXiv preprint arXiv:1709.10441*, 2017.

[Cho and Saul, 2009] Youngmin Cho and Lawrence K Saul. Kernel methods for deep learning. In *Advances in neural information processing systems*, pages 342–350, 2009.

[Cutajar *et al.*, 2017] Kurt Cutajar, Edwin V Bonilla, Pietro Michiardi, and Maurizio Filippone. Random feature expansions for deep gaussian processes. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 884–893. JMLR. org, 2017.

[Delalleau and Bengio, 2011] Olivier Delalleau and Yoshua Bengio. Shallow vs. deep sum-product networks. In *Advances in Neural Information Processing Systems*, pages 666–674, 2011.

[Gibbs, 1998] Mark N Gibbs. *Bayesian Gaussian Processes for Regression and Classification*. PhD thesis, University of Cambridge, 1998.

[Gikhman and Skorohod, 1974] Iosif Ilyich Gikhman and Anatolli Vladimirovich Skorohod. *The Theory of Stochastic Processes*, volume 1. Springer Verlag, Berlin, 1974.

[Heinonen *et al.*, 2016] Markus Heinonen, Henrik Mannerström, Juho Rousu, Samuel Kaski, and Harri Lähdesmäki. Non-stationary gaussian process regression with hamiltonian monte carlo. In *Artificial Intelligence and Statistics*, pages 732–740, 2016.

[Lázaro-Gredilla *et al.*, 2010] Miguel Lázaro-Gredilla, Joaquin Quiñonero-Candela, Carl Edward Rasmussen, and Aníbal R Figueiras-Vidal. Sparse spectrum gaussian process regression. *Journal of Machine Learning Research*, 11:1865–1881, 2010.

[LeCun *et al.*, 1998] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[Rahimi and Recht, 2008] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, pages 1177–1184, 2008.

[Rasmussen and Williams, 2006] Carl Edward Rasmussen and Christopher K I Williams. *Gaussian Process for Machine Learning*. MIT Press, 2006.

[Rebai *et al.*, 2016] Ilyes Rebai, Yassine BenAyed, and Walid Mahdi. Deep multilayer multiple kernel learning. *Neural Computing and Applications*, 27(8):2305–2314, 2016.

[Remes *et al.*, 2017] Sami Remes, Markus Heinonen, and Samuel Kaski. Non-stationary spectral kernels. In *Advances in Neural Information Processing Systems*, pages 4642–4651, 2017.

[Sampson and Guttorp, 1992] Paul D. Sampson and Peter Guttorp. Nonparametric estimation of nonstationary spatial covariance structure. *Publications of the American Statistical Association*, 87(417):108–119, 1992.

[Stein, 1999] Michael L Stein. *Interpolation of Spatial Data*. Springer New York, 1999.

[Sun *et al.*, 2018] Shengyang Sun, Guodong Zhang, Chaoqi Wang, Wenyuan Zeng, Jiaman Li, and Roger Grosse. Differentiable compositional kernel learning for gaussian processes. *arXiv preprint arXiv:1806.04326*, 2018.

[Ton *et al.*, 2018] Jean Francois Ton, Seth Flaxman, Dino Sejdinovic, and Samir Bhatt. Spatial mapping with gaussian processes and nonstationary fourier features. *Spatial Statistics*, 2018.

[Wilson and Adams, 2013] Andrew Gordon Wilson and Ryan Prescott Adams. Gaussian process kernels for pattern discovery and extrapolation. In *International Conference on Machine Learning*, pages 1067–1075, 2013.

[Wilson and Nickisch, 2015] Andrew Wilson and Hannes Nickisch. Kernel interpolation for scalable structured gaussian processes (kiss-gp). In *International Conference on Machine Learning*, pages 1775–1784, 2015.

[Wilson *et al.*, 2016a] Andrew G Wilson, Zhiting Hu, Ruslan R Salakhutdinov, and Eric P Xing. Stochastic variational deep kernel learning. In *Advances in Neural Information Processing Systems*, pages 2586–2594, 2016.

[Wilson *et al.*, 2016b] Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel learning. In *Artificial Intelligence and Statistics*, pages 370–378, 2016.

[Xu *et al.*, 2017] Hai-Ming Xu, Hui Xue, Xiao-Hong Chen, and Yun-Yun Wang. Solving indefinite kernel support vector machine with difference of convex functions programming. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[Yaglom, 1987] Akira Moiseevich Yaglom. *Correlation Theory of Stationary and Related Random Functions*, volume 1. Springer Series in Statistics, 1987.

[Zhang *et al.*, 2017] Shuai Zhang, Jianxin Li, Pengtao Xie, Yingchun Zhang, Minglai Shao, Haoyi Zhou, and Mengyi Yan. Stacked kernel network. *arXiv preprint arXiv:1711.09219*, 2017.

[Zhuang *et al.*, 2011] Jinfeng Zhuang, Ivor W Tsang, and Steven CH Hoi. Two-layer multiple kernel learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 909–917, 2011.