BaKer-Nets: Bayesian Random Kernel Mapping Networks

Hui Xue[1,2*] and **Zheng-Fan Wu**[1,2]
[1]School of Computer Science and Engineering, Southeast University, Nanjing, 210096, China
[2]Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, Nanjing, 210096, China
{hxue, zfwu}@seu.edu.cn

Abstract

Recently, deep spectral kernel networks (DSKNs) have attracted wide attention. They consist of periodic computational elements that can be activated across the whole feature spaces. In theory, DSKNs have the potential to reveal input-dependent and long-range characteristics, and thus are expected to perform more competitive than prevailing networks. But in practice, they are still unable to achieve the desired effects. The structural superiority of DSKNs comes at the cost of the difficult optimization. The periodicity of computational elements leads to many poor and dense local minima in loss landscapes. DSKNs are more likely stuck in these local minima, and perform worse than expected. Hence, in this paper, we propose the novel Bayesian random Kernel mapping Networks (BaKer-Nets) with preferable learning processes by escaping randomly from most local minima. Specifically, BaKer-Nets consist of two core components: 1) a prior-posterior bridge is derived to enable the uncertainty of computational elements reasonably; 2) a Bayesian learning paradigm is presented to optimize the prior-posterior bridge efficiently. With the well-tuned uncertainty, BaKer-Nets can not only explore more potential solutions to avoid local minima, but also exploit these ensemble solutions to strengthen their robustness. Systematical experiments demonstrate the significance of BaKer-Nets in improving learning processes on the premise of preserving the structural superiority.

1 Introduction

With the rapid development of machine learning, most classic kernels are no longer suitable for solving increasingly complex problems. Actually, some studies have figured out that there are two fundamental drawbacks: 1) the inefficiency in computational elements; 2) the limitation on locality [Bengio *et al.*, 2007a; Bengio *et al.*, 2007b]. Concretely, the inefficiency means that the representation ability of these kernels depends heavily on exponentially sufficient computational elements [Delalleau and Bengio, 2011], and the locality means stationarity and monotonicity [Bengio *et al.*, 2006]. In response to such a situation, a new kind of more competitive kernels termed as deep kernels have been arising.

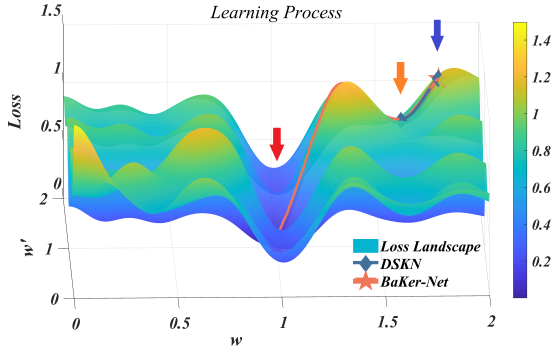


Figure 1: The learning processes of DSKN and BaKer-Net.

Coveting the superiority of deep architectures in representation ability, most deep kernels directly combine deep neural networks as the front-end or the back-end of classic kernels. Typically, the deep kernel learning algorithms set feedforward neural networks as the front-end of spectral mixture kernels to extract features [Wilson *et al.*, 2016b], which are subsequently improved by kernel interpolation [Wilson and Nickisch, 2015] and stochastic variational inference [Wilson *et al.*, 2016a]. Neural kernel networks use sum-product networks as the back-end of multiple kernels to merge various mappings [Sun *et al.*, 2018]. Ironically, as the cask effect implies, the unsolved locality limits the combined kernels and further interferes with the whole networks seriously.

Therefore, deep spectral kernel networks (DSKNs) have been presented to not only improve the efficiency but also break the locality at a stroke [Xue *et al.*, 2019]. They derive non-stationary and non-monotonic kernel mappings to avoid the limitation on locality. These powerful mappings are essentially periodic computational elements that can be activated across the whole feature spaces. Consequently, DSKNs have the potential to reveal input-dependent and long-range characteristics, and thus are expected to perform better than prevailing kernels and deep neural networks. But as yet, DSKNs are still unable to achieve the desired effects in prac-

[*]Contact Author

tical applications. They are inclined to make over-confident but inaccurate decisions, due to low-quality optimization.

In fact, the reason for such a performance bottleneck is that the structural superiority of DSKNs comes at the cost of the profoundly hard optimization. Although the periodicity of computational elements enables non-stationarity and non-monotonicity, it also increases the complexity of networks, and gives rise to numerous poor and dense local minima in loss landscapes. Yet, this fundamental issue has not been taken into account with deliberation. DSKNs are more likely stuck in local minima, and perform worse than expected. Hence, it is necessary to alleviate the difficulty of optimization on the premise of preserving the structural superiority.

Specifically, in this paper, we propose the novel Bayesian random Kernel mapping Networks (BaKer-Nets) in the light of the definite motivation, that is to improve the learning processes by escaping from most poor and dense local minima with some probability. The key is to enable the proper uncertainty of computational elements, which is implemented by two core components:

- To enable the uncertainty of computational elements reasonably, a prior-posterior bridge with copulas is derived.
- To optimize the prior-posterior bridge efficiently, a Bayesian learning paradigm with stochastic variational inference is presented.

Hence, with the well-tuned uncertainty, the advantages of BaKer-Nets depend on two aspects:

- More potential solutions can be explored to avoid poor local minima in learning processes.
- These ensemble solutions can be exploited to strengthen their robustness in generalization processes.

To illustrate this issue intuitively, we conduct a synthetic experiment to learn 1-dimensional $\cos(x)$, where $x \in [-4\pi, 4\pi]$. Both DSKN and BaKer-Net have only one computational element including two weights ω, ω'. As shown in Figure 1, the loss landscape is rugged and rough even though the target function is simple enough. DSKN and BaKer-Net are initialized at the same point marked by the blue arrow. DSKN is stuck quickly in the poor local minimum marked by the yellow arrow. In contrast, BaKer-Net escapes from the local minimum and achieves a much better solution marked by the red arrow, as the optimization continues.

Systematical experiments further demonstrate the competitive performance of BaKer-Nets, and indicate the significance in improving the learning processes on the premise of preserving the structural superiority.

2 Preliminary

According to Yaglom's theorem [Yaglom, 1987], a real-valued bounded continuous function k on $\mathbb{R}^D \times \mathbb{R}^D$ is a non-local positive semi-definite kernel with non-stationarity and non-monotonicity, if it can be represented as

$$k(\boldsymbol{x}, \boldsymbol{x}') = C_+ \int \mathcal{E}_{\boldsymbol{\omega},\boldsymbol{\omega}'}(\boldsymbol{x}, \boldsymbol{x}') p(\boldsymbol{\omega}, \boldsymbol{\omega}') d\boldsymbol{\omega} d\boldsymbol{\omega}', \tag{1}$$

where p is a probability density associated to some probability distribution P, and C_+ is a non-negative scaling constant.

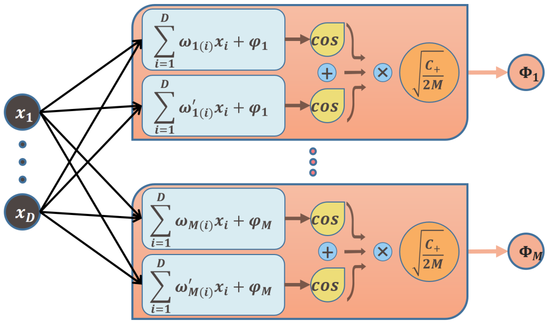

x_1
x_D
$\sum_{i=1}^{D} \omega_{1(i)} x_i + \varphi_1$
$\sum_{i=1}^{D} \omega'_{1(i)} x_i + \varphi_1$
$\sum_{i=1}^{D} \omega_{M(i)} x_i + \varphi_M$
$\sum_{i=1}^{D} \omega'_{M(i)} x_i + \varphi_M$
cos
cos
cos
cos
+
×
+
×
$\sqrt{\frac{C_+}{2M}}$
$\sqrt{\frac{C_+}{2M}}$
Φ_1
Φ_M

Figure 2: The structure of the kernel mapping Φ.

$\mathcal{E}_{\boldsymbol{\omega},\boldsymbol{\omega}'}(\boldsymbol{x}, \boldsymbol{x}')$ is defined by

$$\begin{aligned}
&\frac{1}{8}\Big[e^{i(\boldsymbol{\omega}^T\boldsymbol{x}-\boldsymbol{\omega}'^T\boldsymbol{x}')} + e^{i(-\boldsymbol{\omega}^T\boldsymbol{x}+\boldsymbol{\omega}'^T\boldsymbol{x}')} \\
&+e^{i(\boldsymbol{\omega}'^T\boldsymbol{x}-\boldsymbol{\omega}^T\boldsymbol{x}')} + e^{i(-\boldsymbol{\omega}'^T\boldsymbol{x}+\boldsymbol{\omega}^T\boldsymbol{x}')} \\
&+e^{i(\boldsymbol{\omega}^T\boldsymbol{x}-\boldsymbol{\omega}^T\boldsymbol{x}')} + e^{i(-\boldsymbol{\omega}^T\boldsymbol{x}+\boldsymbol{\omega}^T\boldsymbol{x}')} \\
&+e^{i(\boldsymbol{\omega}'^T\boldsymbol{x}-\boldsymbol{\omega}'^T\boldsymbol{x}')} + e^{i(-\boldsymbol{\omega}'^T\boldsymbol{x}+\boldsymbol{\omega}'^T\boldsymbol{x}')}\Big].
\end{aligned} \tag{2}$$

Furthermore, Eq. (1) can be equivalently transformed into

$$k(\boldsymbol{x}, \boldsymbol{x}') = C_+ \int \mathcal{T}_{\boldsymbol{\omega},\boldsymbol{\omega}'}(\boldsymbol{x}, \boldsymbol{x}') p(\boldsymbol{\omega}, \boldsymbol{\omega}') d\boldsymbol{\omega} d\boldsymbol{\omega}', \tag{3}$$

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

$$\begin{aligned}
\frac{1}{2}\mathbb{E}_{\varphi\sim[-\pi,\pi]}\Big[&\cos(\boldsymbol{\omega}^T\boldsymbol{x}+\varphi)\cos(\boldsymbol{\omega}'^T\boldsymbol{x}'+\varphi) \\
&+\cos(\boldsymbol{\omega}'^T\boldsymbol{x}+\varphi)\cos(\boldsymbol{\omega}^T\boldsymbol{x}'+\varphi) \\
&+\cos(\boldsymbol{\omega}^T\boldsymbol{x}+\varphi)\cos(\boldsymbol{\omega}^T\boldsymbol{x}'+\varphi) \\
&+\cos(\boldsymbol{\omega}'^T\boldsymbol{x}+\varphi)\cos(\boldsymbol{\omega}'^T\boldsymbol{x}'+\varphi)\Big].
\end{aligned} \tag{4}$$

Obviously, Eq. (3) is an expectation on $(\boldsymbol{\omega}, \boldsymbol{\omega}') \sim P$ and $\varphi \sim [-\pi, \pi]$, and thus it can be approximated unbiasedly by Monte Carlo method.

$$k(\boldsymbol{x}, \boldsymbol{x}') = C_+\mathbb{E}_{(\boldsymbol{\omega},\boldsymbol{\omega}')\sim P}\Big[\mathcal{T}_{\boldsymbol{\omega},\boldsymbol{\omega}'}(\boldsymbol{x}, \boldsymbol{x}')\Big] \approx \langle\Phi(\boldsymbol{x}), \Phi(\boldsymbol{x}')\rangle, \tag{5}$$

where $\Phi(\boldsymbol{x})$ is defined by

$$\Phi(\boldsymbol{x}) = \sqrt{\frac{C_+}{2M}} \begin{bmatrix} \cos(\boldsymbol{\omega}_1^T\boldsymbol{x}+\varphi_1) + \cos(\boldsymbol{\omega}_1'^T\boldsymbol{x}+\varphi_1) \\ \vdots \\ \cos(\boldsymbol{\omega}_M^T\boldsymbol{x}+\varphi_M) + \cos(\boldsymbol{\omega}_M'^T\boldsymbol{x}+\varphi_M) \end{bmatrix}_M . \tag{6}$$

D, M are the dimensions of inputs and weights, respectively.

At this point, the non-stationary and non-monotonic kernel mapping Φ is derived. The detailed structure of Φ is illustrated in Figure 2. Moreover, DSKNs are constructed by naturally integrating these specially-designed kernel mappings into deep architectures layer-by-layer. By the way, stacked random Fourier features can be considered as the stationary cases of DSKNs to some extent [Zhang *et al.*, 2017].

According to Eq. (6) and Figure 2, Φ is actually a double-edged sword that consists of a group of periodic computational elements essentially. On the one hand, the periodicity

Copula	$C_\theta(u,v)$
Ali-Mikhail-Haq	$\frac{uv}{1-\theta(1-u)(1-v)}$
Clayton	$\max\{u^{-\theta}+v^{-\theta}-1,0\}^{-\theta^{-1}}$
Frank	$-\frac{1}{\theta}\log[1+\frac{(e^{-\theta u}-1)(e^{-\theta v}-1)}{e^{-\theta}-1}]$
Gumbel	$e^{-[(-\log u)^\theta+(-\log v)^\theta]^{\theta^{-1}}}$

Table 1: Some important bivariate copulas.

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

Table 2: Some classic kernels and their probability densities.

enables non-stationarity and non-monotonicity to break the limitation on locality. Thus, DSKNs have the potential to efficiently reveal input-dependent characteristics and long-range correlations in theory. On the other hand, the periodicity also causes the extremely high complexity of networks, and leads to numerous poor and dense local minima in loss landscapes. Hence, DSKNs are more likely stuck in these local minima, and perform worse than expected in practice.

But as yet, this fundamental issue still has not been taken into account very well. DSKNs directly optimize all weights $(\boldsymbol{\omega},\boldsymbol{\omega}')$ with point estimation, instead of sampling them from the intrinsic probability distribution P. Therefore, the optimization is directly affected by the periodicity of computational elements. Moreover, the essential uncertainty of computational elements is neglected, and thus DSKNs lose the ability to escape from any local minimum.

To improve the learning processes by escaping randomly from poor and dense local minima, it is necessary to enable the proper uncertainty of computational elements on the premise of preserving the structural superiority.

3 BaKer-Nets

In this section, we elaborate the methodology of BaKer-Nets: 1) a prior-posterior bridge with copulas is derived to enable the uncertainty of computational elements; 2) a Bayesian learning paradigm with stochastic variational inference is presented to optimize the prior-posterior bridge.

3.1 Prior-Posterior Bridges

To enable the uncertainty of computational elements, it needs to derive a probability density $p(\boldsymbol{\omega},\boldsymbol{\omega}')$ and its probability distribution $P(\boldsymbol{\omega},\boldsymbol{\omega}')$, according to Eq. (5). The performance of BaKer-Nets almost completely depends on p and P. Thus, it is very important to construct universal and scalable p and P in an interpretable way.

Compared with aimlessly random initialization with intolerable risks, it is a better choice to reasonably derive the powerful p and P with classic kernels as the prior knowledge. In this situation, these initial classic kernels can be regarded as the special cases of BaKer-Nets under specific constraints. At least, the practical performance of BaKer-Nets is guaranteed to be better than that of classic ones. With proper optimization, BaKer-Nets can achieve more competitive performance.

Copulas are vital to bridge the gap between BaKer-Nets and classic kernels. In more detail, Sklar's theorem states that a multivariate joint probability density can be decomposed into univariate marginal probability densities, univariate marginal probability distributions and a copula density [Sklar, 1959]. Here, we pay attention to deriving the D-dimensional bivariate joint probability density $p(\boldsymbol{\omega},\boldsymbol{\omega}')$ from the D-dimensional univariate marginal ones $\bar{p}(\boldsymbol{\omega}),\bar{p}'(\boldsymbol{\omega}')$.

Specifically, given two stationary classic kernels $\bar{k}(\boldsymbol{x}),\bar{k}'(\boldsymbol{x}')$, their probability densities $\bar{p}(\boldsymbol{\omega}),\bar{p}'(\boldsymbol{\omega}')$ can be derived by

$$\begin{aligned}\bar{p}(\boldsymbol{\omega}) =& C_+\int e^{-i\boldsymbol{\omega}\boldsymbol{x}}\bar{k}(\boldsymbol{x})d\boldsymbol{x},\\ \bar{p}'(\boldsymbol{\omega}') =& C_+\int e^{-i\boldsymbol{\omega}'\boldsymbol{x}'}\bar{k}'(\boldsymbol{x}')d\boldsymbol{x}'.\end{aligned} \tag{7}$$

Furthermore, considering the dimensional consistency between kernels and probability densities, $p(\boldsymbol{\omega},\boldsymbol{\omega}')$ can be modeled by integrating $\bar{p}(\boldsymbol{\omega}),\bar{p}'(\boldsymbol{\omega}')$ with bivariate copula densities $\{c^i\}_{i=1}^D$ for all dimensions $\forall i=1,\cdots,D$.

$$p^i(\omega^i,\omega'^i)=c^i(\bar{P}^i(\omega^i),\bar{P}'^i(\omega'^i))\bar{p}^i(\omega^i)\bar{p}'^i(\omega'^i), \tag{8}$$

where $\bar{P}^i(\omega^i),\bar{P}'^i(\omega'^i)$ are the distributions associated to the densities $\bar{p}^i(\omega^i)\bar{p}'^i(\omega'^i)$, respectively. Without losing generality, the distribution $P^i(\omega^i,\omega'^i)$ with the density $p^i(\omega^i,\omega'^i)$ can be also derived by

$$P^i(\omega^i,\omega'^i)=C^i(\bar{P}^i(\omega^i),\bar{P}'^i(\omega'^i)), \tag{9}$$

where C^i is the copula with the density c^i. Consequently, the density $p(\boldsymbol{\omega},\boldsymbol{\omega}')$ and the distribution $P(\boldsymbol{\omega},\boldsymbol{\omega}')$ are as follows.

$$\begin{aligned}p(\boldsymbol{\omega},\boldsymbol{\omega}') =& \{p^i(\omega^i,\omega'^i)\}_{i=1}^D,\\ P(\boldsymbol{\omega},\boldsymbol{\omega}') =& \{P^i(\omega^i,\omega'^i)\}_{i=1}^D.\end{aligned} \tag{10}$$

$(\boldsymbol{\omega},\boldsymbol{\omega}')$ can be sampled from P dimension-by-dimension.

$$(\boldsymbol{\omega},\boldsymbol{\omega}')=\{(\omega^i,\omega'^i)\}_{i=1}^D, \tag{11}$$

where $(\omega^i,\omega'^i)\sim P^i$.

The whole framework, termed as copula-based prior-posterior bridge, essentially connects the prior $\bar{p},\bar{p}'$ and the posterior p. $\bar{p},\bar{p}'$ represent the characteristics of marginals. $\{c^i\}_{i=1}^D$ model their intricate correlation structures. Therefore, we can construct the universal and scalable posterior p for BaKer-Nets by choosing different classic kernels and copulas. There are many parametric copulas available, such as the Gaussian copula [Li, 2000; Aas *et al.*, 2006] and the Archimedean copula family [Charpentier and Segers, 2009; Whelan, 2004]. Some important bivariate copulas are shown in Table 1. Some representative classic kernels and their probability densities are also collected in Table 2.

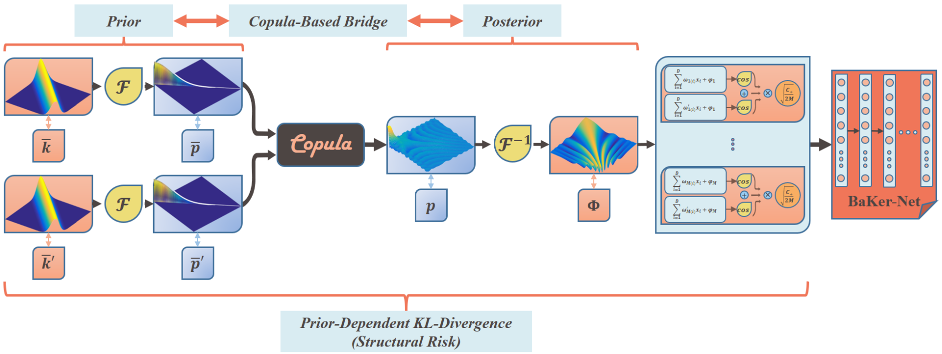


Figure 3: The prior-dependent KL-divergence $KL(p(\boldsymbol{\Omega},\boldsymbol{\Omega}')\|\bar{p}(\boldsymbol{\Omega})\bar{p}'(\boldsymbol{\Omega}'))$.

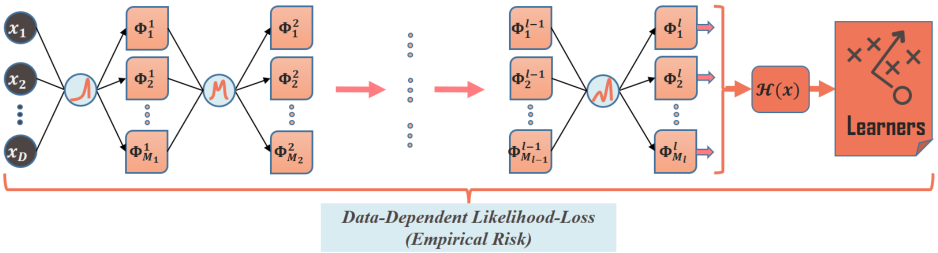


Figure 4: The data-dependent likelihood-loss $-\mathbb{E}_{\boldsymbol{\Omega},\boldsymbol{\Omega}'\sim P}\big[\log\mathbb{P}(\mathcal{D}|\boldsymbol{\Omega},\boldsymbol{\Omega}')\big]$.

3.2 Bayesian Learning Paradigms with Stochastic Variational Inference

Owing to the non-analytically sampling process $(\boldsymbol{\omega},\boldsymbol{\omega}')\sim P$, a Bayesian learning paradigm with stochastic variational inference is further presented to efficiently optimize the whole network, including the prior-posterior bridge.

Above all, the notation of weights in all layers are simplified to $\boldsymbol{\Omega},\boldsymbol{\Omega}'$, that is $(\boldsymbol{\Omega},\boldsymbol{\Omega}') = \{(\boldsymbol{\Omega}^i,\boldsymbol{\Omega}'^i)\}_{i=1}^l$ and $(\boldsymbol{\Omega}^i,\boldsymbol{\Omega}'^i) = \{(\boldsymbol{\omega}_j^i,\boldsymbol{\omega}_j'^i)\}_{j=1}^M$. Thus, the probability density is further represented as

$$p(\boldsymbol{\Omega},\boldsymbol{\Omega}') = \prod_{i=1}^{l}\prod_{j=1}^{M} p^i(\boldsymbol{\omega}_j^i,\boldsymbol{\omega}_j'^i). \tag{12}$$

$\bar{p}(\boldsymbol{\Omega}),\bar{p}'(\boldsymbol{\Omega}')$ have similar independent assumption.

Specifically, given the observed data $\mathcal{D}$ and the weights $\boldsymbol{\Omega},\boldsymbol{\Omega}'$ sampled from the posterior probability density p associated to some prior probability densities $\bar{p},\bar{p}'$, a learning problem can be defined by a log-probability $\log\mathbb{P}(\mathcal{D})$. According to Bayesian inference, it can be formalized as follows.

$$\begin{aligned}\log\mathbb{P}(\mathcal{D}) &= \log\mathbb{P}(\mathcal{D},\boldsymbol{\Omega},\boldsymbol{\Omega}') - \log\mathbb{P}(\boldsymbol{\Omega},\boldsymbol{\Omega}'|\mathcal{D})\\ &= \log\frac{\mathbb{P}(\mathcal{D},\boldsymbol{\Omega},\boldsymbol{\Omega}')}{p(\boldsymbol{\Omega},\boldsymbol{\Omega}')} - \log\frac{\mathbb{P}(\boldsymbol{\Omega},\boldsymbol{\Omega}'|\mathcal{D})}{p(\boldsymbol{\Omega},\boldsymbol{\Omega}')}\\ &= \underbrace{\int\log\frac{\mathbb{P}(\mathcal{D},\boldsymbol{\Omega},\boldsymbol{\Omega}')}{p(\boldsymbol{\Omega},\boldsymbol{\Omega}')}p(\boldsymbol{\Omega},\boldsymbol{\Omega}')d\boldsymbol{\Omega}d\boldsymbol{\Omega}'}_{\mathcal{L}(\mathcal{D},\boldsymbol{\Omega},\boldsymbol{\Omega}')}\\ &+\underbrace{\int\log\frac{p(\boldsymbol{\Omega},\boldsymbol{\Omega}')}{\mathbb{P}(\boldsymbol{\Omega},\boldsymbol{\Omega}'|\mathcal{D})}p(\boldsymbol{\Omega},\boldsymbol{\Omega}')d\boldsymbol{\Omega}d\boldsymbol{\Omega}'}_{KL(p(\boldsymbol{\Omega},\boldsymbol{\Omega}')\|\mathbb{P}(\boldsymbol{\Omega},\boldsymbol{\Omega}'|\mathcal{D}))}.\end{aligned} \tag{13}$$

Because $\log\mathbb{P}(\mathcal{D})$ is a constant when $\mathcal{D}$ is given, maximizing the Evidence Lower BOund (ELBO) $\mathcal{L}(\mathcal{D},\boldsymbol{\Omega},\boldsymbol{\Omega}')$ is equivalent to minimizing the Kullback–Leibler divergence (KL-divergence) $KL(p(\boldsymbol{\Omega},\boldsymbol{\Omega}')\|\mathbb{P}(\boldsymbol{\Omega},\boldsymbol{\Omega}'|\mathcal{D}))$. Generally, we consider optimizing $\mathcal{L}(\mathcal{D},\boldsymbol{\Omega},\boldsymbol{\Omega}')$ with all available parameters $\boldsymbol{\theta}$ including the parameters of the copula-based prior-posterior bridge.

$$\begin{aligned}&\arg\min_{\boldsymbol{\theta}} -\mathcal{L}(\mathcal{D},\boldsymbol{\Omega},\boldsymbol{\Omega}')\\ &=\arg\min_{\boldsymbol{\theta}}\Big[\underbrace{\int\log\frac{p(\boldsymbol{\Omega},\boldsymbol{\Omega}')}{\bar{p}(\boldsymbol{\Omega})\bar{p}'(\boldsymbol{\Omega}')}p(\boldsymbol{\Omega},\boldsymbol{\Omega}')d\boldsymbol{\Omega}d\boldsymbol{\Omega}'}_{KL-Divergence}\\ &\qquad\underbrace{-\int\log\mathbb{P}(\mathcal{D}|\boldsymbol{\Omega},\boldsymbol{\Omega}')p(\boldsymbol{\Omega},\boldsymbol{\Omega}')d\boldsymbol{\Omega}d\boldsymbol{\Omega}'}_{Likelihood-Loss}\Big]\\ &=\arg\min_{\boldsymbol{\theta}}\Big[\underbrace{KL(p(\boldsymbol{\Omega},\boldsymbol{\Omega}')\|\bar{p}(\boldsymbol{\Omega})\bar{p}'(\boldsymbol{\Omega}'))}_{Structural\ Risk}\\ &\qquad\underbrace{-\mathbb{E}_{\boldsymbol{\Omega},\boldsymbol{\Omega}'\sim P}\big[\log\mathbb{P}(\mathcal{D}|\boldsymbol{\Omega},\boldsymbol{\Omega}')\big]}_{Empirical\ Risk}\Big].\end{aligned} \tag{14}$$

Then, the optimization problem $-\mathcal{L}(\mathcal{D},\boldsymbol{\Omega},\boldsymbol{\Omega}')$ can be directly solved by Monte Carlo method.

$$\begin{aligned}&\arg\min_{\boldsymbol{\theta}} -\mathcal{L}(\mathcal{D},\boldsymbol{\Omega},\boldsymbol{\Omega}')\\ &\approx\arg\min_{\boldsymbol{\theta}}\Big[\frac{1}{K}\sum_{i=1}^{K}\big[\log p(\boldsymbol{\Omega},\boldsymbol{\Omega}') - \log(\bar{p}(\boldsymbol{\Omega})\bar{p}'(\boldsymbol{\Omega}'))\big]\\ &\qquad-\frac{1}{K}\sum_{i=1}^{K}\log\mathbb{P}(\mathcal{D}|\boldsymbol{\Omega},\boldsymbol{\Omega}')\Big],\end{aligned} \tag{15}$$

where K is the number of sampling the weights $\mathbf{\Omega}, \mathbf{\Omega}'$. The analytical gradient $\nabla_{\boldsymbol{\theta}}\big[-\mathcal{L}(\mathcal{D}, \mathbf{\Omega}, \mathbf{\Omega}')\big]$ is approximated unbiasedly by [Ranganath *et al.*, 2014; Mandt and Blei, 2014]

$$\begin{aligned}&\nabla_{\boldsymbol{\theta}}\big[-\mathcal{L}(\mathcal{D}, \mathbf{\Omega}, \mathbf{\Omega}')\big]\\ \approx&\frac{1}{K}\sum_{i=1}^{K}\nabla_{\boldsymbol{\theta}}\log p(\mathbf{\Omega}, \mathbf{\Omega}')\big[\log p(\mathbf{\Omega}, \mathbf{\Omega}') - \log(\bar{p}(\mathbf{\Omega})\bar{p}'(\mathbf{\Omega}'))\big]\\ &-\frac{1}{K}\sum_{i=1}^{K}\nabla_{\boldsymbol{\theta}}\log p(\mathbf{\Omega}, \mathbf{\Omega}')\log \mathbb{P}(\mathcal{D}|\mathbf{\Omega}, \mathbf{\Omega}').\end{aligned} \tag{16}$$

Rao-Blackwellization method [Casella and Robert, 1996] and control variates [Ross, 1994] can be further used to reduce the variance of the gradient estimator.

According to the above optimization, the copula-based prior-posterior bridge is not directly interfered by the periodicity of computational elements. BaKer-Nets focus on optimizing the parameters $\boldsymbol{\theta}$ of the prior-posterior bridge rather than the internal weights $\mathbf{\Omega}, \mathbf{\Omega}'$. Thus, the loss landscapes of optimizing BaKer-Nets are much flatter than that of DSKNs, and the the intrinsic structural superiority is preserved very well. Based on the improved learning processes, BaKer-Nets can escape from some poor and dense local minima with proper probability, and thus can achieve better performance and stability. All components in BaKer-Nets can be jointly optimized by prevailing algorithms, such as SGD and Adam, with the derived analytical gradient. Besides, either the time complexity (serial sampling) or the space complexity (parallel sampling) of BaKer-Nets is linearly correlated with K.

As shown in Figure 3 and Figure 4, the prior-dependent $KL(p(\mathbf{\Omega}, \mathbf{\Omega}')\|\bar{p}(\mathbf{\Omega})\bar{p}'(\mathbf{\Omega}'))$ represents the complexity and the potential structural risk, which indicates that it is prone to be simple and generalizable by taking the network closer to the initial classic kernels $\bar{k}, \bar{k}'$. Correspondingly, the data-dependent $-\mathbb{E}_{\mathbf{\Omega}, \mathbf{\Omega}' \sim P}\big[\log \mathbb{P}(\mathcal{D}|\mathbf{\Omega}, \mathbf{\Omega}')\big]$ reflects the learning ability and the practical empirical risk, which indicates that it is inclined to be complex and powerful by revealing highly non-linear and highly-varying details implied in the observed data $\mathcal{D}$. Therefore, the complexity dominated by the KL-divergence and the learning ability dominated by the likelihood-loss strike an elegant balance in BaKer-Nets.

In addition, unlike the conventional point estimation in DSKNs, the weights $\mathbf{\Omega}, \mathbf{\Omega}'$ here are represented as random variables with proper uncertainty subject to the posterior probability distribution P. Thus instead of learning a single network, the proposed approach learns an infinite ensemble of networks in the sense of probability, where each network has its weights drawn from the shared P. The ensemble with greater uncertainty leads naturally to better exploration and exploitation. More potential solutions can be explored to avoid poor local minima in learning processes. These ensemble solutions can be exploited to enhance their robustness in generalization processes.

4 Experiments

In this section, we systematically evaluate the practical performance of BaKer-Nets compared with state-of-the-art related algorithms.

4.1 Experimental Settings

For standardization, we cautiously follow the experimental settings in the official publication of DSKNs [Xue *et al.*, 2019]. Specifically, the scales of all deep architectures are set to $1000 \times 500 \times 50$. Sigmoid activation is applied to deep neural networks. Moreover, all algorithms are initialized according to the Xavier method [Glorot and Bengio, 2010], and are optimized by Adam [Kingma and Ba, 2014]. The learning rate is initially set to a commonly-used default value 0.001 [Paszke *et al.*, 2017], which is automatically tuned by the optimizer. Epochs are set to be large enough to ensure the convergence for all algorithms. Accuracy and Mean Squared Error (MSE) are chosen as the evaluation criteria for classification and regression, respectively.

To be representative, the well-known Gaussian kernels are used as the initial classic kernels $\bar{k}, \bar{k}'$. Thus, the corresponding prior probability densities $\bar{p}, \bar{p}'$ are Gaussian probability densities. Their intricate correlation structures are modeled by a group of bivariate Gaussian copulas.

Compared Algorithms

BaKer-Net is compared with related algorithms including:

- **DNN** [Goodfellow *et al.*, 2016]: Deep Neural Networks.
- **DKL-LI** [Wilson *et al.*, 2016b]: Deep Kernel Learning with LInear kernels.
- **DKL-GA** [Wilson *et al.*, 2016b]: Deep Kernel Learning with GAussian kernels.
- **DKL-SM** [Wilson *et al.*, 2016b]: Deep Kernel Learning with Spectral Mixture kernels.
- **SRFF** [Zhang *et al.*, 2017]: Stacked Random Fourier Features.
- **DSKN** [Xue *et al.*, 2019]: Deep Spectral Kernel Networks.

Datasets

Firstly, we conduct a benchmark experiment on four classification datasets and four regression datasets, which are collected from UCI [Blake and Merz, 1998] and LIBSVM [Chang and Lin, 2011]. These data are randomly divided into non-overlapping training and test sets, which are equal in size. The division, training and test are independently repeated ten times. We assess the convergent performance on average. Secondly, we conduct an image classification experiment on *MNIST*, *FMNIST* and *CIFAR10* [LeCun *et al.*, 1998; Xiao *et al.*, 2017; Krizhevsky *et al.*, 2009], and analyze the learning processes. The division of image datasets is consistent with their default settings. Here, all deep architectures follow the classic design in LeNet [LeCun *et al.*, 1998].

4.2 Experimental Results

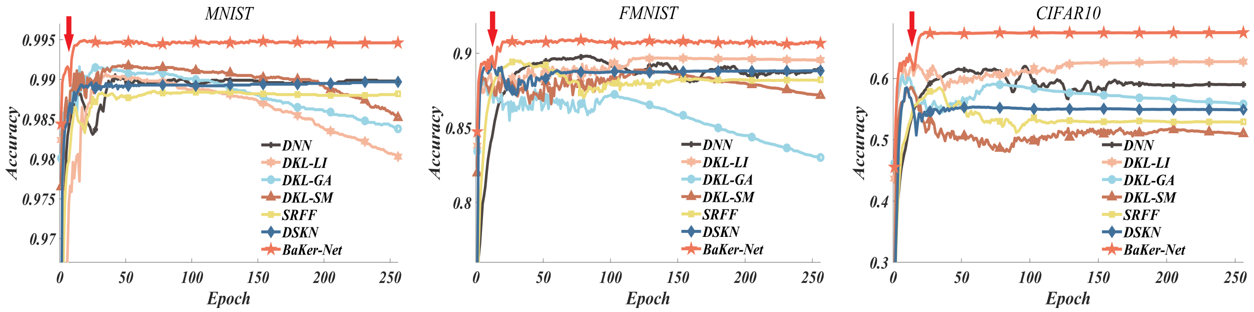
Benchmark

To evaluate the comprehensive performance of BaKer-Nets, the benchmark experiment is conducted.

As shown in Table 3, although these deep kernel learning algorithms are based on DNN, they have relatively poor performance. In these algorithms, the combined kernels more likely interfere with the feature extraction, due to their unsolved locality. Whether in classification or regression, the performance of DSKN is similar to that of SRFF. The structural superiority of DSKN is wasted to some extent. In contrast, BaKer-Net impressively outperforms these compared

	Classification Accuracy (↑)				Regression MSE (↓)				
	a3a	*ionosphere*	*sonar*	*wbdc*	*airfoil*	*power*	*wine-red*	*wine-white*	**R.**
DNN	0.806±0.024●	0.828±0.103●	0.783±0.126	0.933±0.107	0.080±0.010●	0.056±0.002●	0.662±0.036●	0.649±0.011●	**(4)(6)**
DKL-LI	0.818±0.010●	0.809±0.118●	0.658±0.110●	0.976±0.006●	0.078±0.005●	0.059±0.001●	0.629±0.025	0.635±0.019	**(5)(2)**
DKL-GA	0.816±0.010●	0.743±0.115●	0.605±0.112●	0.902±0.148	0.117±0.039●	0.059±0.002●	0.623±0.020	0.634±0.008●	**(7)(5)**
DKL-SM	0.819±0.009●	0.788±0.106●	0.652±0.118●	0.940±0.103	0.144±0.019●	0.062±0.004●	0.651±0.020●	0.657±0.027●	**(6)(7)**
SRFF	0.802±0.006●	0.882±0.020●	0.818±0.039●	0.961±0.009●	0.076±0.011●	0.061±0.001●	0.631±0.023	0.638±0.017●	**(3)(3)**
DSKN	0.818±0.011●	0.917±0.033	0.819±0.038●	0.974±0.007●	0.063±0.010●	0.055±0.002●	0.652±0.030●	0.651±0.012●	**(2)(4)**
BaKer-Net	**0.835±0.008**	**0.934±0.022**	**0.859±0.040**	**0.983±0.003**	**0.051±0.002**	**0.046±0.001**	**0.612±0.020**	**0.622±0.005**	**(1)(1)**

Table 3: Classification accuracy and regression MSE (mean±std.) on the benchmark datasets. (↑) indicates the larger the better, while (↓) indicates the smaller the better. The best results are highlighted in **bold** and the average ranks on accuracy and MSE are listed in **R.**. ●/○ indicates whether BaKer-Net is statistically superior/inferior to the compared algorithms (pairwise *t*-test at 0.05 significance level).



(a) *MNIST* (b) *FMNIST* (c) *CIFAR10*

Figure 5: Test accuracy curves in all epochs. Different curves represent the learning processes of different algorithms where BaKer-Net is denoted as the best orange-red curve.

	MNIST		*FMNIST*		*CIFAR10*	
	conv	**best**	**conv**	**best**	**conv**	**best**
DNN	0.989	0.990	0.889	0.898	0.590	0.620
DKL-LI	0.980	0.990	0.895	0.896	0.627	0.627
DKL-GA	0.983	0.991	0.830	0.883	0.558	0.605
DKL-SM	0.985	0.991	0.871	0.888	0.510	0.585
SRFF	0.988	0.988	0.882	0.894	0.529	0.585
DSKN	0.989	0.989	0.888	0.893	0.549	0.585
BaKer-Net	**0.995**	**0.995**	**0.908**	**0.911**	**0.674**	**0.675**

Table 4: Classification accuracy on the image datasets. **conv** means the convergent accuracy in the last epoch and **best** means the best accuracy in all epochs. The best results are highlighted in **bold**.

algorithms on all datasets. The results explicitly demonstrate that BaKer-Net is compatible with practical learning tasks well, and achieves credible performance improvement.

Image Classification

Specially, to demonstrate that BaKer-Net can improve the learning processes by escaping from some poor and dense local minima, the image classification experiment is conducted. The results are illustrated in Table 4 and Figure 5.

With the increasing difficulty of *MNIST*, *FMNIST* and *CIFAR10*, almost all compared algorithms gradually fall into worse and worse fluctuations. The trends are clearly presented in Figure 5. In the early stages of optimization, these compared algorithms fall into annoying local minima, and begin to fluctuate in $0 - 50$ epochs, as indicated by the red arrows. As the optimization continues, they are still stuck in these poor local minima. After the automatically-tuned learning rates almost decrease to 0, they converge to terrible solutions. In contrast, due to enabling the uncertainty of computational elements, BaKer-Net escapes from these local minima and further achieves better performance by exploring more potential solutions. It also enhances robustness by exploiting these ensemble solutions. Consequently, BaKer-Net not only achieves the best accuracy but also obtains great stability, benefiting from the prior-posterior bridge and the Bayesian learning paradigm.

5 Conclusion

To alleviate the difficulty of optimization on the premise of preserving the structural superiority, we propose BaKer-Nets with preferable learning processes. Specifically, a prior-posterior bridge is derived to enable the uncertainty of computational elements reasonably. Subsequently, a Bayesian learning paradigm is presented to optimize the prior-posterior bridge efficiently. Hence, BaKer-Nets can not only explore more potential solutions to avoid local minima, but also exploit these ensemble solutions to strengthen their robustness. Systematical experiments demonstrate the competitive performance of the proposed approach, and further indicate the significance of BaKer-Nets in improving learning processes.

Acknowledgments

This work was supported by the National Key R&D Program of China (2017YFB1002801). It was also supported by Collaborative Innovation Center of Wireless Communications Technology.

References

[Aas *et al.*, 2006] Kjersti Aas, Claudia Czado, Arnoldo Frigessi, and Henrik Bakken. Pair-copula constructions of multiple dependence. *Insurance Mathematics & Economics*, 44(2):182–198, 2006.

[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.*, 2007a] 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.

[Bengio *et al.*, 2007b] Yoshua Bengio, Yann LeCun, et al. Scaling learning algorithms towards ai. *Large-scale kernel machines*, 34(5):1–41, 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.

[Casella and Robert, 1996] George Casella and Christian P Robert. Rao-blackwellisation of sampling schemes. *Biometrika*, 83(1):81–94, 1996.

[Chang and Lin, 2011] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[Charpentier and Segers, 2009] Arthur Charpentier and Johan Segers. Tails of multivariate archimedean copulas. *Journal of Multivariate Analysis*, 100(7):1521–1537, 2009.

[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.

[Glorot and Bengio, 2010] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[Goodfellow *et al.*, 2016] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT press, 2016.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Krizhevsky *et al.*, 2009] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[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.

[Li, 2000] David X Li. On default correlation: A copula function approach. *Social Science Electronic Publishing*, 9(4), 2000.

[Mandt and Blei, 2014] Stephan Mandt and David Blei. Smoothed gradients for stochastic variational inference. In *Advances in Neural Information Processing Systems*, pages 2438–2446, 2014.

[Paszke *et al.*, 2017] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[Ranganath *et al.*, 2014] Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Artificial Intelligence and Statistics*, pages 814–822, 2014.

[Ross, 1994] Sheldon M Ross. A new simulation estimator of system reliability. *International Journal of Stochastic Analysis*, 7(3):331–336, 1994.

[Sklar, 1959] M Sklar. Fonctions de répartition À n dimensions et leurs marges. *Publ.inst.statist.univ, Paris*, 8:229–231, 1959.

[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.

[Whelan, 2004] Niall Whelan. Sampling from archimedean copulas. *Quantitative Finance*, 4(3):339–352, 2004.

[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.

[Xiao *et al.*, 2017] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[Xue *et al.*, 2019] Hui Xue, Zheng-Fan Wu, and Wei-Xiang Sun. Deep spectral kernel learning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 4019–4025. International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[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.