DeepKriging: Spatially Dependent Deep Neural Networks for Spatial Prediction

Yuxiao Li[1], Ying Sun[1], Brian J Reich [2]

July 28, 2020

Abstract

In spatial statistics, a common objective is to predict the values of a spatial process at unobserved locations by exploiting spatial dependence. In geostatistics, Kriging provides the best linear unbiased predictor using covariance functions and is often associated with Gaussian processes. However, when considering non-linear prediction for non-Gaussian and categorical data, the Kriging prediction is not necessarily optimal, and the associated variance is often overly optimistic. We propose to use deep neural networks (DNNs) for spatial prediction. Although DNNs are widely used for general classification and prediction, they have not been studied thoroughly for data with spatial dependence. In this work, we propose a novel neural network structure for spatial prediction by adding an embedding layer of spatial coordinates with basis functions. We show in theory that the proposed DeepKriging method has multiple advantages over Kriging and classical DNNs only with spatial coordinates as features. We also provide density prediction for uncertainty quantification without any distributional assumption and apply the method to $PM_{2.5}$ concentrations across the continental United States.

Keywords: Deep learning, Gaussian processes, Radial basis function, Spatial Regression, Feature embedding

[1] Statistics Program, King Abdullah University of Science and Technology, Thuwal 23955-6900, Saudi Arabia. E-mail: yuxiao.li@kaust.edu.sa; ying.sun@kaust.edu.sa

[2] North Carolina State University, Department of Statistics, Campus Box 8203, 5212 SAS Hall, Raleigh, NC 27695. Email: brian_reich@ncsu.edu

This research is supported by King Abdullah University of Science and Technology (KAUST), Office of Sponsored Research (OSR) under Award No: OSR-2019-CRG7- 3800.

1 Introduction

Spatial prediction is at the heart of spatial and spatio-temporal statistics. Common objectives are to predict a spatial process at unobserved locations and to study the spatial dependence in the region of interest. An ideal spatial prediction not only provides point prediction, but also distributional information such as quantiles or the density function to quantify uncertainties, risks and extreme values (Diggle et al., 2007). Traditional applications of spatial prediction are in the fields of geological and environmental science (Cressie, 2015), but they have been extended to other fields, such as biological sciences, computer vision, economics and public health (Anselin, 2001; Austin, 2002; Waller and Gotway, 2004; Franchi et al., 2018).

The primary collection of spatial prediction methods are based on best linear unbiased prediction (BLUP), also referred to as Kriging (Matheron, 1963). Kriging prediction is a weighted average of observed data points, where the weights are determined by the spatial covariance or variogram of the random process. Under the Gaussian assumption, Kriging also provides the full predictive distribution. Applying Kriging requires estimating the spatial covariance function. To make the estimation problem tractable, it is common to assume the covariance function is stationary, i.e., is the same throughout the entire spatial domain.

However, physical processes tend to be non-Gaussian and non-stationary. For instance, the data on wind speed and fine particles ($PM_{2.5}$) exposures are positive, right-skewed, and sometimes heavy-tailed (Hennessey Jr, 1977; Adgate et al., 2002) and likely the spatial covariance varies across space, e.g., in urban versus rural areas (Sampson et al., 2013). It is possible to derive the best linear prediction for certain parametric non-Gaussian processes (Xu and Genton, 2017; Rimstad and Omre, 2014) and certain non-stationary covariance structures (Fuentes, 2002; Paciorek and Schervish, 2004; Li and Sun, 2019). However, spatial prediction for more general spatial processes remains an open problem.

Another drawback of the Kriging prediction is the computational cost, which is prohibitive for large spatial datasets because the Kriging prediction involves the inversion of an $N \times N$ positive definite covariance matrix, where N is the number of observed locations (Heaton et al., 2019). The calculation is typically done by the Cholesky decomposition which requires $O(N^3)$ time and $O(N^2)$ memory complexity.

Recently, deep learning or deep neural networks (DNNs) have become the most powerful prediction tools for a wide range of applications, especially in computer vision and natural language processing (LeCun et al., 2015). DNNs are effective for predicting with complex features such as non-linearity and non-stationarity, and computationally efficient in analyzing massive datasets using GPUs (Najafabadi et al., 2015). However, there are two major obstacles encountered when applying deep learning to spatial predictions. First, classical deep neural networks (DNNs) cannot incorporate spatial dependence directly. Applications in spatial prediction with neural networks usually simply include spatial coordinates as features (Cracknell and Reading, 2014), which may not be sufficient. Recently, convolutional neural networks (CNNs, Krizhevsky et al. 2012) have been stated to successfully capture the spatial and temporal dependencies in image processing through the relevant filters. However, the framework is designed for applications with a large feature space,

and often requires large training labels as the ground truth, which does not fit for many spatial prediction problems, where only in-situ and sparse observations are available. Compared to CNNs, the goal of spatial prediction is to account for spatial correlations in the response variable with limited observed features and spatially sparse observations.

Second, traditional DNNs cannot provide the uncertainty information at the predicted spatial locations. Recently, several methods have been proposed to overcome this problem by predicting the entire probabilistic distribution. For example, Li et al. (2019) discretized the density function of the response conditional on covariates as a histogram in a regression model, and estimated the density by using neural networks to classify predicted values into different bins. Neal (2012), Gal and Ghahramani (2016) and Posch et al. (2019) applied Bayesian inference methodologies to neural networks to predict uncertainties via the posterior distribution. However, these methods cannot be applied directly to spatial data. Few studies exist on embedding neural networks into spatial prediction in geostatistics. To our knowledge, only Wang et al. (2019) proposed a nearest-neighbor neural network for spatial processes by considering local Kriging predictors as features to account for neighboring information. However, they did not provide a general framework for spatial prediction and uncertainty quantifications with neural networks.

Therefore, motivated by the Karhunen–Loève theorem (Adler, 2010), we propose a spatially dependent neural network by adding an embedding layer of spatial coordinates using basis functions. The proposed method is suitable for non-Gaussian or categorical data and contributes to the spatial prediction in at least five aspects:

1) it builds a direct link between neural networks and Kriging in the spatial prediction;

2) it models spatial dependence through a set of basis functions rather than covariance functions and allows for non-stationarity;

3) it does not require the inversion of the covariance matrix and is scalable for massive datasets;

4) it provides a non-linear predictor in covariates and generally in observations;

5) it measures the uncertainty through predictive density functions without assuming any data distribution.

We call our method "DeepKriging" with the aim of achieving the optimal spatial prediction, similarly to the original use of Kriging (Cressie, 1990), but by deep neural networks. The proposed framework is different from classical Kriging methods that construct the optimal linear predictor based on a certain covariance function or variogram. However, using the Bayesian learning theory (Neal, 2012; Lee et al., 2017), we show in theory that the Deepkriging model can approximate Gaussian processes with a rich class of covariance functions. We also show that our model is superior to Kriging in terms of minimizing the approximation error (see details in Section 4). Due to modern deep learning platforms such as Keras (Gulli and Pal, 2017), our method is easy to implement and can be accelerated using GPUs. We apply the approach to $PM_{2.5}$ concentration data across the continental United States, and show that DeepKriging outperforms Kriging based on cross-validation.

The rest of our paper is organized as follows. Section 2 introduces the general framework of spatial prediction and Kriging. Section 3 introduces the proposed DeepKriging methods including

its uncertainty quantification. Section 4 provides the properties and theoretical foundation of DeepKriging. Section 5 presents some simulation studies to show the performance of DeepKriging. Section 6 applies our method to predict $PM_{2.5}$ concentration in the U.S.. Section 7 summarizes our main results and suggests directions for future work.

2 Spatial Prediction and Kriging

We propose a general framework for both Kriging and neural networks to perform spatial prediction. Consider a spatial process $\{Y(\mathbf{s}) : \mathbf{s} \in D\}, D \subseteq \mathbb{R}^d$, to be the real-valued spatial process of interest, and let $\mathbf{Z} = \{Z(\mathbf{s}_1), \ldots, Z(\mathbf{s}_N)\}^T$ be the measurements that we observed at N locations.

The goal of spatial prediction is to find the best predictor $\widehat{Y}^{\text{opt}}(\mathbf{s}_0, \mathbf{Z})$ of the true process at an unobserved location $\mathbf{s}_0$, as a function of $\mathbf{Z}$. In decision theory, $\widehat{Y}^{\text{opt}}(\mathbf{s}_0, \mathbf{Z})$ is viewed as the minimizer of an expected loss function or risk function (DeGroot, 2005). That is,

$$\widehat{Y}^{\text{opt}}(\mathbf{s}_0, \mathbf{Z}) = \underset{\hat{Y}}{\text{argmin}}\, \mathbb{E}\{L(\widehat{Y}(\mathbf{s}_0), Y(\mathbf{s}_0))\} = \underset{\hat{Y}}{\text{argmin}}\, R(\widehat{Y}(\mathbf{s}_0), Y(\mathbf{s}_0)), \tag{1}$$

where $L(\cdot, \cdot)$ is a loss function and $R(\cdot, \cdot)$ is a risk function. The loss function is chosen by the type of observations and the class of targeted predictors. For continuous data, if we are interested in the mean, median, and quantile prediction, then the mean squared loss, mean absolute loss, and check loss could be used, respectively. For other types of data such as categorical or count data, related loss functions are also available and have been reviewed by Zhao et al. (2015). Take the most widely used loss function, the mean squared error (MSE) as an example. Under the MSE loss, if the mean and variance of the error is finite, then the minimum mean square error (MMSE) predictor is $\widehat{Y}^{\text{opt}}(\mathbf{s}_0, \mathbf{Z}) = \mathbb{E}\{Y(\mathbf{s}_0)|\mathbf{Z}\}$. The MMSE predictor has multiple good properties such as unbiasedness and asymptotic normality under the regularity assumptions (Lehmann and Casella, 2006). In particular, if $Y(\mathbf{s}_0)$ and $\mathbf{Z}$ are jointly Gaussian, the conditional mean has a closed form, and the MMSE predictor is linear in observations $\mathbf{Z}$. If $Y(\mathbf{s}_0)$ and $\mathbf{Z}$ are not jointly Gaussian, the conditional mean obtained by Gaussian assumption remains the best linear unbiased prediction (BLUP).

Kriging (Matheron, 1963) is the method to find the BLUP in spatial prediction and often referred to as Gaussian process regression under the Gaussian assumption. Consider a generalized additive model, where we observe $\mathbf{Z}$ from a spatial process defined as $Z(\mathbf{s})$. It assumes that $Z(\mathbf{s}) = Y(\mathbf{s}) + \varepsilon(\mathbf{s})$, where $\varepsilon(\mathbf{s})$ is a white noise process, called the nugget effect, with zero mean and variance $\sigma(\mathbf{s})^2$, caused by measurement inaccuracy and fine-scale variability. The spatial process $Y(\mathbf{s})$ is typically assumed to be $Y(\mathbf{s}) = \mathbf{x}(\mathbf{s})^T\boldsymbol{\beta} + \nu(\mathbf{s})$, where $\mathbf{x}(\mathbf{s}) \in \mathbb{R}^P$ is a vector process of P known covariates, $\boldsymbol{\beta}$ is a vector of coefficients, $\nu(\mathbf{s})$ is a spatially dependent and zero-mean random process with a generally non-stationary covariance function, $\mathbb{C}\text{ov}(\nu(\mathbf{s}_i), \nu(\mathbf{s}_j)) = C(\mathbf{s}_i, \mathbf{s}_j)$. Let $\delta(\mathbf{s}) = \nu(\mathbf{s}) + \varepsilon(\mathbf{s})$, then for $\boldsymbol{\delta} = \{\delta(\mathbf{s}_1), \ldots, \delta(\mathbf{s}_N)\}^T$, we have $\mathbb{E}(\boldsymbol{\delta}) = 0$ and $\mathbb{C}\text{ov}(\boldsymbol{\delta}) = \boldsymbol{\Sigma}$, where the (i, j)-th element is $\boldsymbol{\Sigma}_{i,j} = C(\mathbf{s}_i, \mathbf{s}_j) + \sigma(\mathbf{s}_i)^2\mathbb{1}\{i = j\}$.

Under this model, the (universal) Kriging prediction is

$$\widehat{Y}^{\mathrm{UK}}(\mathbf{s}_0) = \mathbf{x}(\mathbf{s}_0)^T\widehat{\boldsymbol{\beta}} + \mathbf{c}(\mathbf{s}_0)^T\boldsymbol{\Sigma}^{-1}(\mathbf{Z} - \mathbf{X}\widehat{\boldsymbol{\beta}}), \tag{2}$$

where $\mathbf{X} = (\mathbf{x}(\mathbf{s}_1), \ldots, \mathbf{x}(\mathbf{s}_N))^T$ is an $N \times P$ matrix, $\mathbf{c}(\mathbf{s}_0) = (C(\mathbf{s}_0, \mathbf{s}_1), \ldots, C(\mathbf{s}_0, \mathbf{s}_N))^T$, and $\widehat{\boldsymbol{\beta}} = (\mathbf{X}^T\boldsymbol{\Sigma}^{-1}\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{\Sigma}^{-1}\mathbf{Z}$. The Kriging variance is the MSE of $\widehat{Y}(\mathbf{s}_0)$, which is

$$\sigma^2_{\mathrm{UK}}(\mathbf{s}_0) = C(\mathbf{s}_0, \mathbf{s}_0) - \mathbf{c}(\mathbf{s}_0)^T\boldsymbol{\Sigma}\mathbf{c}(\mathbf{s}_0) + (\mathbf{x}(\mathbf{s}_0) - \mathbf{X}^T\mathbf{c}(\mathbf{s}_0))^T(\mathbf{X}^T\boldsymbol{\Sigma}^{-1}\mathbf{X})^{-1}(\mathbf{x}(\mathbf{s}_0) - \mathbf{X}^T\mathbf{c}(\mathbf{s}_0)). \tag{3}$$

In practice, the covariance matrix $\boldsymbol{\Sigma}$ in (2) and (3) is unknown and needs to be estimated from data. Typically, the covariance function $C(\mathbf{s}_i, \mathbf{s}_j)$ is assumed to have a parametric form, e.g., exponential or Matérn covariance function (Cressie, 2015), with unknown parameters $\boldsymbol{\theta}$. Then we can fit the parametric model to the empirical covariance/variogram by least square estimation or to the data by a likelihood-based method with certain distribution assumption (Cressie, 2015).

Although Kriging can provide the BLUP and gives the MMSE predictor under the Gaussian assumption, it has several limitations. First, data in many real applications are non-Gaussian and even non-continuous. Hence, minimizing the MSE is not reasonable and the prediction of Kriging will be sub-optimal. Second, even for Gaussian processes, maximum likelihood estimation (MLE) of $\boldsymbol{\theta}$ is computationally expensive. The multivariate Gaussian likelihood involves the inverse of an $N \times N$ positive definite covariance matrix, $\boldsymbol{\Sigma}(\boldsymbol{\theta})$. The calculation of $\boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta})$ generally requires Cholesky decomposition with $O(N^3)$ time and $O(N^2)$ memory complexity which makes Kriging computational infeasible for massive datasets. Therefore, real applications call for more accurate and computationally efficient methods for spatial prediction and uncertainty quantification.

3 DeepKriging

3.1 Deep learning in spatial prediction

We propose to use deep learning for the spatial prediction. The idea is to approximate the optimal predictor $\widehat{Y}^{\mathrm{opt}}(\mathbf{s}_0, \mathbf{Z})$ in (1) by the output with the structure of neural networks. Then the optimal neural network predictor is $f^{\mathrm{opt}}_{\mathrm{NN}}(\mathbf{s}_0) = \mathrm{argmin}_{f_{\mathrm{NN}}} R\{f_{\mathrm{NN}}(\mathbf{s}_0), Y(\mathbf{s}_0)\}$, where $f_{\mathrm{NN}}(\cdot) \in \mathcal{F}$ can be any function in the function space $\mathcal{F}$ expressible by a family of neural networks and $f^{\mathrm{opt}}_{\mathrm{NN}}(\cdot)$ is the best function in $\mathcal{F}$ in terms of minimizing a certain risk $R(\cdot, \cdot)$. The inputs of $f_{\mathrm{NN}}(\cdot)$ can be covariates $\mathbf{x}(\mathbf{s}_0)$, coordinates $\mathbf{s}_0$, and other variables available at $\mathbf{s}_0$ such as the basis functions proposed in the following section. Typically, we write $f^{NN}(\cdot, \boldsymbol{\theta})$ as a parametric model with unknown parameters $\boldsymbol{\theta}$, including the weights and biases in the neural networks.

Note that the optimal neural network predictor $f^{\mathrm{opt}}_{\mathrm{NN}}(\mathbf{x}(\mathbf{s}_0))$ is practically unreachable since $Y(\mathbf{s}_0)$ is unknown. Thus, we approximate the predictor by minimizing the empirical loss function over the training set $\mathbf{Z}$ (Goodfellow et al., 2016). That is, the final predictor of a neural network is $\widehat{Y}_{NN}(\mathbf{s}_0, \mathbf{Z}) = f^{NN}(\mathbf{s}_0, \widehat{\boldsymbol{\theta}})$ and

$$\widehat{\boldsymbol{\theta}} = \mathrm{argmin}_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N} L\{f^{NN}(\mathbf{s}_n, \boldsymbol{\theta}), Z(\mathbf{s}_n)\}, \tag{4}$$

where $\frac{1}{N}\sum_{n=1}^{N} L\{f^{NN}(\mathbf{s}_n, \boldsymbol{\theta}), Z(\mathbf{s}_n)\}$ is the empirical version of the risk $R\{f_{\text{NN}}(\mathbf{s}_0), Y(\mathbf{s}_0)\}$. However, spatial prediction differs from the classical neural network prediction at least in two aspects. First, spatial prediction typically has limited observed features rather than excessive features in common applications of neural networks, such as computer vision. Therefore, the advantage of neural networks in auto feature selection is less relevant. Second, spatial prediction $Y(\mathbf{s})$ typically possesses certain spatial dependence. For example, in geostatistics, we usually follow Tobler's first law of geography (Tobler, 1970): "near things are more related than distant things". Then, rather than simply considering the relationship between $Y(\mathbf{s})$ and $\mathbf{x}(\mathbf{s})$, we are also interested in the spatial dependence in $Y(\mathbf{s})$.

These problems can be further explained by the spatial models. To apply neural networks or other machine learning models, we apply the assumption that $Y(s)|\mathbf{x}(\mathbf{s})$ are mutually independent conditional on features. However, this assumption is not reasonable in spatial prediction because we typically assume that $Y(\mathbf{s}) = \mathbf{x}(\mathbf{s})^T\boldsymbol{\beta} + \nu(\mathbf{s})$, where the covariates $\mathbf{x}(\mathbf{s})$ only contribute to the mean structure of $Y(\mathbf{s})$ and $\nu(\mathbf{s})$ remains a spatially correlated process. Here, more features apart from $\mathbf{x}(\mathbf{s})$ are needed to model the spatial dependence in applying the neural networks.

To account for the spatial information, the most natural way is to add d coordinates (e.g., longitude and latitude) to the features, in the hope that the neural networks can learn the dependent term $\nu(\mathbf{s})$ as a function of $\mathbf{s}$. By doing that, the adjusted features become $\mathbf{x}^{adj}(\mathbf{s}) = (\mathbf{x}(\mathbf{s})^T, \mathbf{s})^T$. However, it may not be efficient with the finite width and depth of a neural network. Although neural networks can potentially approximate any function, it is a composite function of linear functions and simple activation functions. It may take huge effort to achieve the a good approximation if the true function is far from linear. For example, the Kriging predictor is linear in $\mathbf{x}(\mathbf{s})$ but obviously non-linear in coordinates $\mathbf{s}_0$. A more detailed illustration is provided in Section 5.

By reviewing the Kriging predictor in (2), we can see that the spatial dependence is typically incorporated via the covariance vector $\mathbf{c}(\mathbf{s}_0)$, i.e., $\widehat{Y}^{\text{UK}}(\mathbf{s}_0)$ is a linear function of $\mathbf{c}(\mathbf{s}_0)$, where $\mathbf{c}(\mathbf{s}_0)$ consists of N covariances at $\mathbf{s}_0$. Motivated by the covariance vector $\mathbf{c}(\mathbf{s}_0)$ in the Kriging predictor, we propose to use a set of known nonlinear functions as the embedding of $(\mathbf{s})$ in the features. In this way, the final predictor has a simpler relationship with features than adding coordinates directly to the features.

The proposal of embedding $(\mathbf{s})$ in the features by spatial basis functions can be further justified by the Karhunen–Loève (KL) theorem (Adler, 2010) applied to the zero-mean spatial process $\nu(\mathbf{s})$. The theorem establishes that $\nu(\mathbf{s})$ admits a decomposition $\nu(\mathbf{s}) = \sum_{k=1}^{\infty} w_k\phi_k(\mathbf{s})$, where w_k's are pairwise uncorrelated random variables and $\phi_k(\mathbf{s})$'s are pairwise orthogonal basis functions in the domain of $\nu(\mathbf{s})$. Hence, $\nu(\mathbf{s})$ can be linearly quantified by the orthogonal basis functions and uncorrelated random variables.

As its empirical version, the prediction of $\nu(\mathbf{s})$ is typically the truncated KL expansion. As one of its properties, given any orthonormal basis functions $\phi_k(\mathbf{s})$, we can find some integer K, so that $\nu(\mathbf{s})$ can be approximated by the finite weighted sum of basis functions, i.e., $\widehat{\nu}(\mathbf{s}) = \sum_{k=1}^{K} w_k\phi_k(\mathbf{s})$, in the sense of minimizing the total mean square error. Several choices of basis functions have been proposed, such as smoothing spline basis functions (Wahba, 1990), wavelet basis functions

(Vidakovic, 2009), and radial basis functions (Friedman et al., 2001).

Therefore, rather than only including the spatial coordinates $\mathbf{s}$ in the features, we transform the d coordinates to K basis functions. More precisely, we consider the d to K embedding layer in the network before we pass the data to the hidden layers.

3.2 DeepKriging: a spatially dependent neural network

We use a simple DNN to illustrate our DeepKriging framework. The key contribution of Deep Kriging is the novel way of embedding spatial coordinates in the features. Thus, using similar idea, our model can be potentially used in other deep learning frameworks such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs).

First, for any coordinates $\mathbf{s}$, we compute the K basis functions to get the embedded vectors $\boldsymbol{\phi}(\mathbf{s}) = (\phi_1(\mathbf{s}), \ldots, \phi_K(\mathbf{s}))^T$. The basis functions are not necessary but recommended to be orthogonal to follow the K-L expansion. Then let $\mathbf{x}_\phi(\mathbf{s}) = (\mathbf{x}(\mathbf{s})^T, \boldsymbol{\phi}(\mathbf{s})^T)^T$ be the embedded input of length $P + K$, we can specify a L-layer DNN as

$$\begin{aligned}
&\mathbf{u}_1(\mathbf{s}) = \mathbf{W}_1\mathbf{x}_\phi(\mathbf{s}) + \mathbf{b}_1, \;\; \mathbf{a}_1(\mathbf{s}) = \psi_1(\mathbf{u}_1(\mathbf{s}));\\
&\mathbf{u}_2(\mathbf{s}) = \mathbf{W}_2\mathbf{a}_1(\mathbf{s}) + \mathbf{b}_2, \;\; \mathbf{a}_2(\mathbf{s}) = \psi_2(\mathbf{u}_2(\mathbf{s}));\\
&\ldots\\
&\mathbf{u}_L(\mathbf{s}) = \mathbf{W}_L\mathbf{a}_{L-1}(\mathbf{s}) + \mathbf{b}_L, \;\; f^{\mathrm{DK}}(\mathbf{s}) = \psi_L(\mathbf{u}_L(\mathbf{s})).
\end{aligned} \tag{5}$$

For the l-th layer with L_l neurons, $\mathbf{W}_l$ is the $L_l \times L_{l-1}$ weight matrix, $\mathbf{b}_l$ is the bias vector of length L_l, $\mathbf{a}_l$ is the neuron vector of length L_l, $\psi_l(\cdot)$ is the activation function, and $f^{\mathrm{DK}}(\mathbf{s})$ is the output of the DeepKriging model.

Let $\boldsymbol{\theta}$ be those unknown weights and biases and $\widehat{\boldsymbol{\theta}}$ be the estimates by training samples and minimizing the loss function of the neural network as defined in (4). Then the DeepKriging predictor for an unobserved location $\mathbf{s}_0$ is defined as $\widehat{Y}_{DK}(\mathbf{s}_0) = f^{DK}(\mathbf{s}_0, \widehat{\boldsymbol{\theta}})$

One major advantage of DeepKriging is that we can adjust the number of neurons, activation functions, and loss functions to fit for different data types and model interpretations. For example, for predicting continuous variables as in a regression problem, we choose $L_L = 1$, $\psi_L(\cdot)$ to be an identity function, and the loss to be mean squared error. Figure 1 provides the visualization of DeepKriging in two dimensional prediction for continuous data. For predicting categorical variables as in a classification problem, we choose L_L to be the number of categories, $\psi_L(\cdot)$ to be a softmax function, and the loss to be the cross entropy loss. For the activation functions in the hidden layers, we choose the rectified linear unit (ReLU) as the default setting, which allows us to keep the linear relationship in the KL expansion but add some deactivated neurons to select the best number of basis functions.

The DeepKriging structure also allows for covariate effects to be spatially varying, as well as the distribution prediction and uncertainty quantification. The details and algorithms by applying the deep distribution regression (Li et al., 2019) are shown in the supplementary materials. Other details of the DeepKriging model are also discussed in the supplementary materials, including

choosing the basis function, regularization, and optimization.

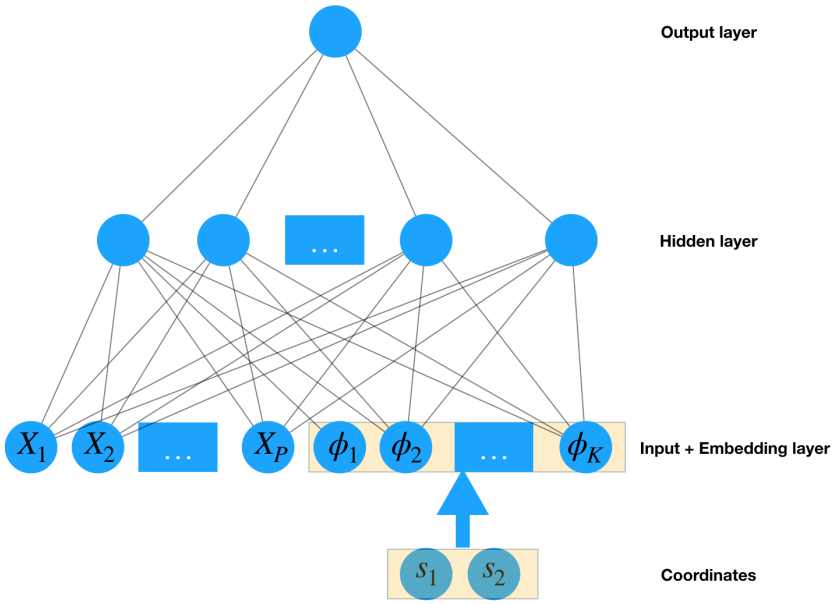

Figure 1: Visualization of the DeepKriging structure in 2D spatial prediction based on a three-layer DNN

4 Properties and Theoretical Foundation of DeepKriging

DeepKriging provides a novel spatial prediction framework using deep learning. It differs from classical Kriging methods in several aspects. First, DeepKriging does not directly model its prediction as a linear combination of observations in Kriging. In contrast, DeepKriging prediction is linked to the observations via its weights and biases through the model training and will be typically nonlinear in observations (see the example in the supplementary materials). Second, DeepKriging does not assume a Gaussian process with a certain covariance function but models spatial dependence by basis functions. Last, unlike Kriging which predicts the random process $Y(\mathbf{s})$ at an unobserved location, DeepKriging approximates the process using a deterministic continuous function.

However, Kriging can provide an optimal prediction under Gaussian assumption and the BLUP in general, and it also provides a tractable way to account for the spatial dependence of random processes by covariance functions. Therefore, several questions are theoretically important to answer about DeepKriging: 1) What is the underlying relationship between DeepKriging and Kriging; 2) How accurate can Deepkriging be in terms of the prediction error compared to Kriging; 3) How to measure the spatial dependence and model the spatial process $Y(\mathbf{s})$ from the DeepKriging framework? The three questions are critical for understanding DeepKriging and we will answer them in the following subsections, respectively.

4.1 The link between DeepKriging and Kriging-based methods

First, DeepKriging is closely related to Kriging and the associated variants. The most direct example is fixed rank Kriging (FRK) proposed by Cressie and Johannesson (2008), which uses one of the low-rank approximations of the covariance matrix in order to speed up the computation of universal Kriging. Similar to DeepKriging, they represent the spatial process by K basis functions, i.e., $\nu(\mathbf{s}) = \boldsymbol{\phi}(\mathbf{s})^T\boldsymbol{\eta}$, where $\boldsymbol{\eta}$ is a K dimensional Gaussian random vector with $\mathbb{C}\text{ov}(\boldsymbol{\eta}) = \boldsymbol{\Sigma}_K$. They call the model for $\nu(\mathbf{s})$ a spatial random effects model and the model for $Y(\mathbf{s}) = \mathbf{x}(\mathbf{s})^T\boldsymbol{\beta} + \boldsymbol{\phi}(\mathbf{s})^T\boldsymbol{\eta}$ a spatial mixed-effects model.

They assume the covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{V} + \boldsymbol{\Phi}\boldsymbol{\Sigma}_K\boldsymbol{\Phi}^T$, where $\boldsymbol{\Phi} = \{\boldsymbol{\phi}(\mathbf{s}_1), \ldots, \boldsymbol{\phi}(\mathbf{s}_N)\}$ is an $N \times K$ basis matrix and $\boldsymbol{V} = \text{diag}\{\sigma(\mathbf{s}_1)^2, \ldots, \sigma(\mathbf{s}_N)^2\}$ is an $N \times N$ diagonal matrix. As a result, the FRK prediction is

$$\widehat{Y}^{\text{FRK}}(\mathbf{s}_0) = \mathbf{x}(\mathbf{s}_0)^T\widehat{\boldsymbol{\beta}} + \boldsymbol{\phi}(\mathbf{s}_0)^T\boldsymbol{\Sigma}_K\boldsymbol{\Phi}^T\boldsymbol{\Sigma}^{-1}(\mathbf{Z} - \mathbf{X}\widehat{\boldsymbol{\beta}}) = \mathbf{x}(\mathbf{s}_0)^T\widehat{\boldsymbol{\beta}} + \boldsymbol{\phi}(\mathbf{s}_0)^T\widehat{\boldsymbol{\alpha}}, \tag{6}$$

where $\mathbf{X} = (\mathbf{x}(\mathbf{s}_1)^T, \cdots, \mathbf{x}(\mathbf{s}_1)^T)^T$ is an $N \times P$ matrix. Equation (6) shows that the FRK prediction $\widehat{Y}^{\text{FRK}}(\mathbf{s}_0)$ is not only a linear function of observations, $\mathbf{Z}$, but also a linear function of P covariates $\mathbf{x}(\mathbf{s}_0)$ and K basis functions $\boldsymbol{\phi}(\mathbf{s}_0)$ at the new location $\mathbf{s}_0$, which become a special case of DeepKriging when we set all of the activation functions to be linear.

However, FRK usually chooses K to be much smaller than N as a computationally efficient method for large datasets. Since the covariance $\boldsymbol{\Phi}\boldsymbol{\Sigma}_K\boldsymbol{\Phi}^T$ has at most rank K, such a low-rank approximation of the covariance matrix may fail to capture the high-frequency variation or small-scale spatial dependence in the spatial process (Stein, 2014).

Other methods with similar ideas are available under either spatial random effects models or K-L expansions. Generally, they can be classified into two classes: multi-resolution processes (Nychka et al., 2015; Kleiber and Nychka, 2015; Katzfuss, 2017) and Gaussian predictive processes (Banerjee et al., 2008, 2010). Although these methods mainly focus on covariance approximations and the estimation is typically based on the likelihood, it is easy to show that the prediction is in fact a linear function of embedded features $\mathbf{x}_\phi(\mathbf{s}_0)$, and thus can be potentially approximated by DeepKriging.

Returning focus to the Kriging itself, although it is less obvious from Equation (2), the Kriging predictor is also related to basis functions. Given that the basis functions at all observed locations form an $N \times K$ basis matrix $\boldsymbol{\Phi}$, Lemma 1 shows that the Kriging predictor with any covariance function can be also expressed by the basis functions when $\boldsymbol{\Phi}$ has rank N.

Lemma 1. *Given that $\boldsymbol{\Phi}$ has rank N and $\nu(\mathbf{s})$ is a zero-mean spatial process with a covariance function $C(\cdot,\cdot)$. The Kriging predictor in (2) is a linear function in $\mathbf{x}_\phi(\mathbf{s}_0)$.*

The detailed proof is provided in the Appendix. It implies that as long as the basis matrix has rank N, a Kriging predictor with any covariance function can be linearly expressed by the embedding layers $\mathbf{x}_\phi(\mathbf{s}_0)$. Therefore, both Kriging and DeepKriging predictions are functions of $\mathbf{x}_\phi(\mathbf{s}_0)$, but DeepKriging generalizes Kriging by allowing for non-linear function of $\mathbf{x}_\phi(\mathbf{s}_0)$.

DeepKriging is also related to a set of non-linear Kriging models, such as indicator Kriging (Journel, 1983; Carr and Mao, 1993) and disjunctive Kriging (Matheron, 1976), in which they

model the indicator and a nonlinear transformation of the response as spatial processes. The traditional methods for indicator Kriging rely on tedious inference and modeling of multiple indicator semivariograms, as well as the post-processing of the results. In contrast, DeepKriging can adjust the output layer to fit for different data types without changing the network structure in the input and hidden layers. Thus, it can be compatible with indictor Kriging and potentially many other tasks with little effort.

4.2 DeepKriging in decision theory

Reviewing the spatial prediction framework in Section 2, both Kriging and DeepKriging aim at achieving the optimal predictor. The prediction procedure conventionally follows an approximation-estimation decomposition as described in Fan et al. (2019). Let $\mathcal{F}$ be the function space expressible by a particular model and $\widehat{Y}_N(\mathbf{s}_0)$ be the final predictor from the model based on N locations. Then we have three types of errors in the prediction.

The first type is the approximation error determined by the function class $\mathcal{F}$ and relates to the capacity of a model. It is defined as the risk between the true process $Y(\mathbf{s}_0)$ and the optimal predictor as a function in $\mathcal{F}$, where the optimal predictor in $\mathcal{F}$ is $\widehat{Y}^{\text{opt}}_{\mathcal{F}}(\mathbf{s}_0) = \underset{\widehat{Y}(\mathbf{s}_0)\in\mathcal{F}}{\text{argmin}}\, R(\widehat{Y}(\mathbf{s}_0), Y(\mathbf{s}_0))$. The second type is the estimation error affected by the complexity of $\mathcal{F}$ and relates to the generalization power of the model, defined as the risk between $\widehat{Y}^{\text{opt}}_N(\mathbf{s}_0)$ and $\widehat{Y}^{\text{opt}}_{\mathcal{F}}(\mathbf{s}_0)$, where $\widehat{Y}^{\text{opt}}_N(\mathbf{s}_0) = \underset{\widehat{Y}_N(\cdot)\in\mathcal{F}}{\text{argmin}} \frac{1}{N}\sum_{n=1}^{N} L\{\widehat{Y}_N(\mathbf{s}_n), Z(\mathbf{s}_n)\}$. The third error is the optimization error, which is the empirical risk between $\widehat{Y}^{\text{opt}}_N(\cdot)$ and $\widehat{Y}_N(\cdot)$ over the training data.

The function class of Kriging, $\mathcal{F}_{UK}$, is the space of linear functions of both $\mathbf{x}(\mathbf{s}_0)$ and $\mathbf{Z}$ taking the form $\mathbf{x}(\mathbf{s}_0)^T\boldsymbol{\beta} + \mathbf{Z}^T\boldsymbol{\alpha}$, while the function class of DeepKriging, $\mathcal{F}_{DK}$, is the function space generated by the DNN described in (5). In fact, with infinite hidden neurons, the space $\mathcal{F}^{DK}$ can be the space of continuous functions on the features $\mathcal{X}_\phi(\mathbf{s}_0)$ defined as $\mathbb{C}(\mathcal{X}_\phi)$, where $\mathcal{X}_\phi(\mathbf{s}_0)$ is the embedded feature space generated by $\mathbf{x}_\phi(\mathbf{s}_0)$. In Lemma 2, we show that the optimal DeepKriging predictor defined in (5) with a sufficient number of hidden neurons and finite loss has the largest capacity in $\mathbb{C}(\mathcal{X}(\mathbf{s}_0))$. The detailed proof is provided in the Appendix.

Lemma 2 (Universal approximation theorem in spatial prediction)**.** *With an arbitrary activation function, the DeepKriging predictor defined in* (5) *with sufficiently many hidden neurons and finite loss satisfies that* $\widehat{Y}^{\text{opt}}_{\mathcal{F}_{DK}}(\mathbf{s}_0) = \widehat{Y}^{\text{opt}}_{\mathbb{C}(\mathcal{X}_\phi)}(\mathbf{s}_0)$.

Theorem 1 follows immediately from Lemmas 1 and 2. It shows DeepKriging has larger capacity than Kriging.

Theorem 1. *Given a set of basis functions and the corresponding basis matrix* $\boldsymbol{\Phi}$ *has rank* N *(*$K \geq N$*), the DeepKriging prediction with a sufficient number of hidden neurons has larger capacity than the Kriging prediction with any covariance function in terms of minimizing the approximation error under certain loss, i.e.,* $\mathbb{E}\{L(\widehat{Y}^{\text{opt}}_{\mathcal{F}_{DK}}(\mathbf{s}_0), Y(\mathbf{s}_0))\} \leq \mathbb{E}\{L(\widehat{Y}^{\text{opt}}_{\mathcal{F}_{UK}}(\mathbf{s}_0), Y(\mathbf{s}_0))\}$.

Proof: From Lemma 1, when $\mathbf{\Phi}$ has rank N, $\mathcal{F}_{UK} \subset \mathbb{C}(\mathcal{X}_\phi)$. Thus from Lemma 2, it shows that $\mathbb{E}\{L(\widehat{Y}^{\text{opt}}_{\mathcal{F}_{DK}}(\mathbf{s}_0), Y(\mathbf{s}_0))\} = \mathbb{E}\{L(\widehat{Y}^{\text{opt}}_{\mathbb{C}(\mathcal{X}_\phi)}(\mathbf{s}_0), Y(\mathbf{s}_0))\} \leq \mathbb{E}\{L(\widehat{Y}^{\text{opt}}_{\mathcal{F}_{UK}}(\mathbf{s}_0), Y(\mathbf{s}_0))\}$. □

To better understand the link between Kriging and DeepKriging, we have the following remarks.

Remark 1. *If $Y(\mathbf{s})$ is a Gaussian process, the DeepKriging predictor with sufficient hidden neurons and rank N basis matrix satisfies that $\widehat{Y}^{\text{opt}}_{\mathcal{F}_{DK}}(\mathbf{s}_0) = \mathbb{E}\{Y(\mathbf{s}_0)|\mathbf{Z}\}$ under MSE.*

Therefore, DeepKriging is more advantageous for model capacity or approximation error than Kriging, when both the number of hidden neurons and the rank of basis matrix are large. However, this does not mean that DeepKriging is superior to Kriging because the estimation error exists. The prediction from finite training samples or observations $\mathbf{Z}$ may not be generalized on the unobserved locations. The generalization issue is considered as the oracle and asymptotic property of the maximum likelihood estimation (MLE) (Fan and Li, 2001) in the fields of Kriging, and as overfitting in the DeepKriging framework. Generally speaking, the model complexity of DeepKriging is much larger than Kriging, which can lead to larger estimation error. However, when the sample size is large, the difference can be significant reduced. The optimization error comes from the numerical optimization in the model training. We do not discuss this part in theory but it can significantly affect the model performance in practice.

In summary, assuming the number of basis functions and hidden layers are sufficiently large, then DeepKriging has a larger model capacity than Kriging, but may be less stable, especially when the size of training data is small.

4.3 DeepKriging as a Gaussian Process

Although DeepKriging does not require the distributional assumption of $Y(\mathbf{s})$, it can induce a Gaussian process (GP) representation similar to Kriging. We follow the Bayesian learning framework proposed by Neal (2012) and Lee et al. (2017) to illustrate the property. As a special case of (5), consider the three-layer and regression-type DeepKriging model with $Y(\mathbf{s}) = b_2 + \sum_{j=1}^{L_1} w_{2j} a_{1j}(\mathbf{s})$, where $a_{1j}(\mathbf{s}) = \psi_1(b_{1j} + \sum_{k=1}^{K} w_{1k} x_{\phi_k}(\mathbf{s}))$. In Bayesian learning, the bias and weights are typically assumed to be $b_2, b_{11}, \ldots, b_{1,L-1} \overset{iid}{\sim} (0, \sigma_b^2)$, $w_{21}, \ldots, w_{2,L_1} \overset{iid}{\sim} (0, \sigma_w^2/L_1)$, and $w_{11}, \ldots, w_{1K} \overset{iid}{\sim} (0, \sigma_w^2)$. Note that $Y(\mathbf{s}_1), \ldots, Y(\mathbf{s}_N)$ is a sum of identically independent random vectors. From the multidimensional central limit theorem, any combination of $Y(\mathbf{s}_1), \ldots, Y(\mathbf{s}_N)$ follows a multivariate Gaussian distribution as L_1 goes to infinity. Therefore, $Y(\mathbf{s})$ is a Gaussian process with zero mean and covariance function $C(\mathbf{s}_i, \mathbf{s}_j) = \mathbb{E}\{Y(\mathbf{s}_i)Y(\mathbf{s}_j)\} = \sigma_b^2 + \sigma_w^2 \mathbb{E}\{a_{11}(\mathbf{s}_i)a_{11}(\mathbf{s}_j)\}$.

For a DNN with more hidden layers, Lee et al. (2017) provides the covariance function in a recursive way, $C^l(\mathbf{s}_i, \mathbf{s}_j) = \sigma_b^2 + \sigma_w^2 F_\psi(C^{l-1}(\mathbf{s}_i, \mathbf{s}_j), C^{l-1}(\mathbf{s}_i, \mathbf{s}_i), C^{l-1}(\mathbf{s}_j, \mathbf{s}_j))$, where $F_\psi(\cdot)$ is a deterministic function that depends only on the activation function ψ. For the base case, $C^0(\mathbf{s}_i, \mathbf{s}_j) = \sigma_b^2 + \sigma_w^2\{\mathbf{x}_\phi(\mathbf{s}_i)\mathbf{x}_\phi(\mathbf{s}_j)/(P+K)\}$. The aforementioned results require the assumption of infinite hidden neurons in each layer. However, when the prior distribution of weights and biases is Gaussian, the limit is not needed.

We start from the most trivial case by letting the activation function in the hidden layers to be an identity function $\psi_l(x) = x$ and $\mathbf{x}(\mathbf{s})$ to be a constant. Then $Y(\mathbf{s})$ becomes a linear function

of the basis functions $\boldsymbol{\phi}(\mathbf{s})$, i.e., $Y(\mathbf{s}) = b_c + \mathbf{w}_c^T\boldsymbol{\phi}(\mathbf{s})$, where b_c and $\mathbf{w}_c$ are combined bias and weights. In this case, the covariance function is $C(\mathbf{s}_i, \mathbf{s}_j) = \sigma_{b_c}^2 + \sigma_{w_c}^2\boldsymbol{\phi}(\mathbf{s}_i)^T\boldsymbol{\phi}(\mathbf{s}_j)$, which is the basis approximation of spatial covariance function. In this sense, DeepKriging can be centered on a parametric spatial process model with the basis functions selected to approximate the spatial covariance function.

When the activation function is not an identity function but ReLU used in the DeepKriging in the hidden layers, we can still have a basis approximation since $\psi(b+x) \approx b+x$ for large b. It also has an explicit form in general proposed by Cho and Saul (2009), which is also the corresponding covariance function of DeepKriging:

$$C^l(\mathbf{s}_i, \mathbf{s}_j) = \sigma_b^2 + \frac{\sigma_w^2}{2\pi}\sqrt{C^{l-1}(\mathbf{s}_i, \mathbf{s}_i)C^{l-1}(\mathbf{s}_j, \mathbf{s}_j)}\left\{\sin\theta_{i,j}^{l-1} + (\pi - \theta_{i,j}^{l-1})\cos\theta_{i,j}^{l-1}\right\}, \tag{7}$$

where $\theta_{i,j}^l = \arccos\left(C^l(\mathbf{s}_i, \mathbf{s}_j)/\sqrt{C^l(\mathbf{s}_i, \mathbf{s}_i)C^l(\mathbf{s}_j, \mathbf{s}_j)}\right)$.

From the results, we can find that DeepKriging also considers the spatial dependence of $\mathbf{x}(\mathbf{s})$ in the covariance function, which is called the covariate-dependent covariance function in the Kriging framework (Reich et al., 2011a; Ingebrigtsen et al., 2014). However, common Kriging methods only assume that the covariance function only depends on $\mathbf{s}$.

The induced covariance function of DeepKriging can also possess favorably physical interpretations. For example, the Matérn covariance function is the most popular choice in Kriging because it relates a stochastic partial differential equation (SPDE) of Laplace type (Whittle, 1954). Likewise, Neal (2012) provided an example of DNN induced GP that can approximate a fractional Brownian motion. The GP representation also implies that the embedding layer in the DeepKriging can bring more flexible spatial covariance structures than simply using the coordinates, which is shown in the Theorem 2.

Theorem 2. *Consider a three-layer and regression-type DeepKriging model without the covariates* $\mathbf{x}(\mathbf{s})$. *The covariance function of* $Y(\mathbf{s})$ *for any two nearby locations has the form* $C(\mathbf{s}_i, \mathbf{s}_j) = v(\mathbf{s}_i) + v(\mathbf{s}_j) - c\|\boldsymbol{\phi}(\mathbf{s}_i) - \boldsymbol{\phi}(\mathbf{s}_j)\|^2$, *where* $\boldsymbol{\phi}(\mathbf{s})$ *is the basis vector at location* $\mathbf{s}$, $v(\mathbf{s}) > 0$ *related to the variance when* $\mathbf{s}_i = \mathbf{s}_j$, *and* c *is the scaling parameter.*

The proof is shown in the Appendix. As its special case, if only the coordinates are used in the features, then $\|\boldsymbol{\phi}(\mathbf{s}_i) - \boldsymbol{\phi}(\mathbf{s}_j)\|^2 = \|\mathbf{s}_i - \mathbf{s}_j\|^2$, $v(\mathbf{s}_i) = v(\mathbf{s}_j) = v$ and $C(\mathbf{s}_i, \mathbf{s}_j) = v - c\|\mathbf{s}_i - \mathbf{s}_j\|^2$.

As an implication of Theorem 2, we can see how the DeepKriging induced covariance function can approximate the common stationary covariance functions in spatial statistics. Let the basis functions be $\phi_l(\mathbf{s}) = k(\mathbf{s}, \mathbf{u}_l)$ from some kernel function k and knot $\mathbf{u}_l$. If the $\mathbf{u}_l$ forms a fine grid of knots covering the spatial domain, then

$$||\phi(\mathbf{s}_i) - \phi(\mathbf{s}_j)||^2 = \sum_{l=1}^{L}\{k(\mathbf{s}_i, \mathbf{u}_l) - k(\mathbf{s}_j, \mathbf{u}_l)\}^2 \approx \int\{k(\mathbf{s}_i, \mathbf{u}) - k(\mathbf{s}_j, \mathbf{u})\}^2 d\mathbf{u}$$
$$= \int k(\mathbf{s}_i, \mathbf{u})^2 + k(\mathbf{s}_j, \mathbf{u})^2 - 2k(\mathbf{s}_i, \mathbf{u})k(\mathbf{s}_j, \mathbf{u})d\mathbf{u}.$$

Note that the third term is the kernel convolution approximation to the covariance function. (Higdon, 2002) shows that by selecting the appropriate kernel function, you can approximate any

stationary spatial covariance function. Therefore, the associated covariance function of DeepKriging is also connected to the common spatial covariance functions.

5 Simulation Studies

5.1 DeepKriging prediction on a Gaussian process

We start from a simple one-dimensional stationary Gaussian process (GP) as our example, where the Kriging prediction is optimal, to validate the DeepKriging method, and to compare to the common practice that only includes the coordinates in DNNs for capturing the spatial dependence for the spatial prediction. The simulated data are generated from a GP with a constant mean, $Z(s) = \mu + \nu(s) + \varepsilon(s)$, where $\mu = 1$, $\nu(s)$ is zero mean GP with an exponential covariance function $C(s_i, s_j)$ with variance $\sigma^2 = 1$ and range parameter $\rho = 0.1$, i.e. $C(s_i, s_j) = \sigma^2 \exp\{-\|s_i - s_j\|/\rho\}$, and $\varepsilon(s)$ is Gaussian white noise with the nugget variance, $\tau^2 = 0.01$. We generate 100 replicates from the Gaussian process with $1,000$ equally spaced locations over $[0, 1]$, among which 800 locations are randomly selected as training data.

In this example, there is no observed covariates except for the intercept, i.e., $x(s) = 1$ for any $s \in \mathbb{R}$. Three different scenarios are considered in the simulation: DNN with $x(s) = 1$ only, DNN with $x(s)$ and coordinate s, and DeepKriging with $x(s)$ and embedding layer. We also compare to a Kriging prediction with the true exponential covariance function and the Kriging prediction with an estimated Matérn covariance function with the smoothness parameter set to 1.5. The predictions related to Gaussian processes and deep learning are implemented using GPy library (GPy, 2012) and Keras library (Ketkar et al., 2017), respectively.

Figure 2 shows the prediction for whole datasets of one sample using the five predictors. It can be seen that DNN with the intercept only predicts the mean of the process. Although including the coordinate s in DNN can provide better results, it fails to capture the high-frequency variability and cannot reflect the spatial correlations of the true process. On the other hand, DeepKriging and Kriging predictors are almost overlapped on both training and testing sets.

To further validate the performance, Table 1 shows the root mean squared error (RMSE) and mean absolute percentage error (MAPE) on the testing set over 100 replicates, where MAPE is defined as $\frac{1}{N_{\text{test}}} \sum_{n=1}^{N_{\text{test}}} \frac{Y_{\text{pred}} - Y_{\text{true}}}{Y_{\text{true}}}$, N_{test} is the number of testing samples, Y_{pred} is the predicted value and Y_{true} is the true value. As the MMSE predictor, the Kriging prediction with the true covariance function has the smallest RMSE. However, we can see that the performance of DeepKriging is comparable to the two Kriging predictors for Gaussian processes and significantly outperforms the classical DNN models. Besides, the MAPE of DeepKriging is smaller than the Kriging with an estimated covariance function. This simulation study shows that even for Gaussian processes, DeepKriging may provide a prediction as good as Kriging.

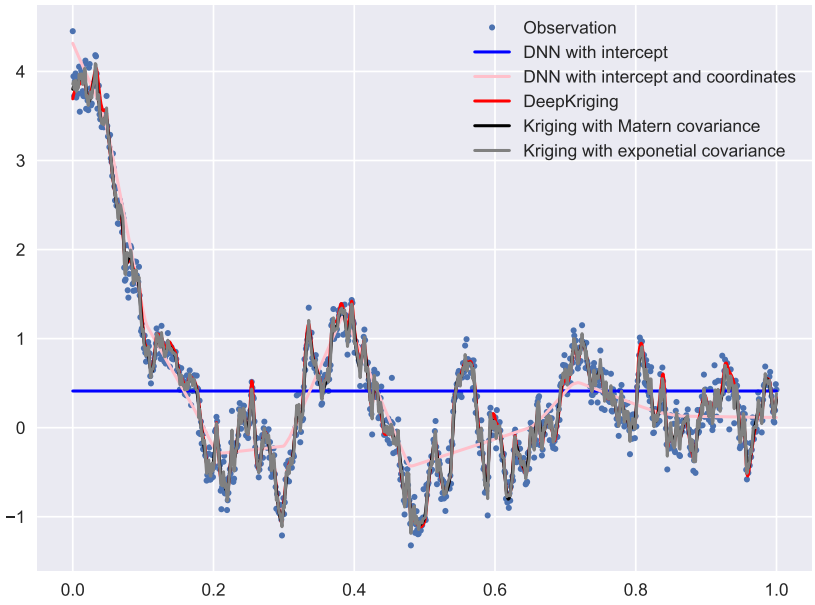

Figure 2: The prediction results for one dataset generated from a Gaussian Process. The blue dots are the simulated samples from the GP simulation. The solid lines are the prediction results from five different models: DNN with intercept only (blue line), DNN with both intercept and coordinates, DeepKriging (red line), Kriging with the true exponential covariance function (grey line), and Kriging with an estimated Matérn covariance function.

Table 1: Root mean squared error (RMSE) and mean absolute percentage error (MAPE) from the five predictions of a Gaussian process. Standard deviations across 100 datasets are given in parentheses. Kriging I and II are the Kriging prediction from a true exponential covariance function and an estimated Matérn covariance function, respectively. DeepKriging is the DeepKriging prediction based on the mean square loss. DNN I and DNN II are the DNN prediction with intercept and with both intercept and coordinates, respectively.

Models	Kriging I		Kriging II		DeepKriging		DNN I		DNN II	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
RMSE	**.160**	.010	.166	.009	.171	.010	.845	.176	.294	.025
MAPE	**.521**	.337	.556	.407	.548	.390	2.955	1.354	.903	.730

5.2 DeepKriging prediction on a function with non-stationary feature

The second example focuses on two-dimensional data with non-stationary feature so that the procedure is designed to resemble the real data in Section 6. The goal is to approximate a computationally expensive simulation with the true function:

$$f(\mathbf{s}) = \sin\{30(\bar{s} - 0.9)^4\}\cos\{2 * (\bar{s} - 0.9)\} + (\bar{s} - 0.9)/2,$$

where $\bar{s} = (s_x + s_y)/2$, and $\mathbf{s} = (s_x, s_y)^T \in \mathbb{R}^2$. A similar example is evaluated in many computer experiments (Ba et al., 2012; Xiong et al., 2007), where both Kriging and neural networks are popularly applied. In our simulation, $N = 900$ observations are sampled on a $30\times$ 30 square grid of locations spanning $[0, 1]^2$ (Figure 3a). The function shows obvious non-stationary features in space since the smoothness in the region of $[0, 0.4]^2$ is significantly smaller than that of $[0.4, 1]^2$. To examine the prediction, we use the 10-fold cross-validation to show the performance of DeepKriging, Kriging and the baseline DNN only with coordinates s. We calculate the mean squared error and mean absolute error, which are shown in the first two rows of Table 2 and Figure 3(b).

DeepKriging significantly outperforms Kriging in the simulation, because the Kriging prediction that assumes a stationary covariance function. In contrast, due to the universal approximation theorem, DeepKriging is expected to approximate any function in 2D, even if it is non-stationarity. On the other hand, a baseline DNN is also better than Kriging in this example because it also satisfies the universal approximation theorem. However, it is worse than DeepKriging for a relatively small depth of neural networks as we have explained in the Section 3.2.

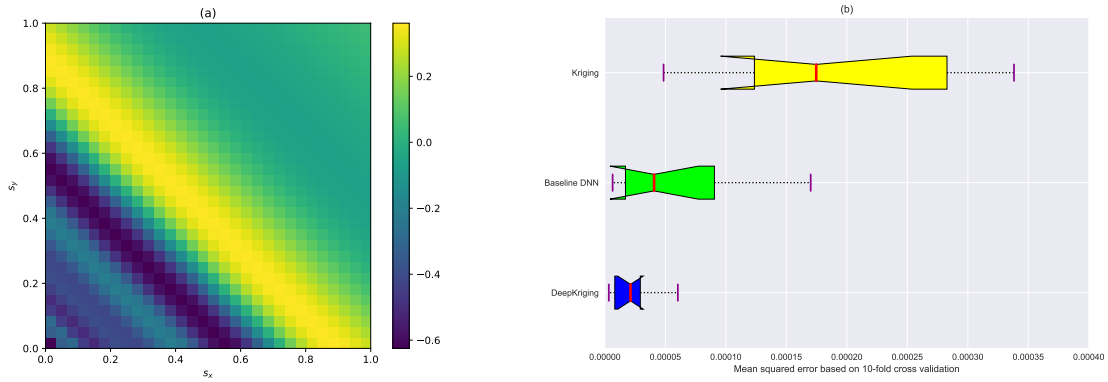

Figure 3: (a) Visualization of the simulated data as an image. The data are generated from $f(s) = \sin\{30(\bar{s}-0.9)^4\}\cos\{2(\bar{s}-0.9)\} + (\bar{s}-0.9)/2$, where $\bar{s} = (s_x + s_y)/2$, and $\mathbf{s} = (s_x, s_y)^T \in [0, 1]^2$. (b) The boxplots of the 10 mean squared errors based on the 10-fold cross-validations from DeepKriging (blue), baseline DNN (green), and Kriging (yellow).

Table 2: The model performance based on the 10-fold cross-validation. We choose the mean squared error (MSE) and mean absolutely error (MAE) as the validation criteria. The results are the mean and standard deviation (SD) of the 10 sets of validation errors.

Parameters	DeepKriging		Baseline DNN		Kriging	
	Mean	SD	Mean	SD	Mean	SD
MSE ($\times 10^{-4}$)	**0.43**	.0.68	0.94	1.32	2.90	3.50
MAE ($\times 10^{-3}$)	**3.66**	2.80	5.14	2.82	6.25	1.58

5.3 DeepKriging is non-linear in observation

Although the DeepKriging predictor is not assumed to be linear in observations in Kriging, it is important to investigate how the predictor is linked with the observations. We designed a simulation with 100 observations generated by $Z(s) = Y(s)\mathbb{1}_{\{Y(s)>0\}} + \varepsilon(s)$, where $Y(s) = 10\cos(20s)$ and $s \in [0,1]$, the coordinates s are regularly located in $[0,1]$, and $\varepsilon(s) \overset{iid}{\sim} N(0,1)$. The simulated data $Z(s)$ and signal $Y(s)$ are shown in Figure 4(a). The figure also shows the prediction results from both Kriging and DeepKriging, which are almost identical.

To examine the non-linear relationship between a training observation and the prediction, we replace the observation $Z(s)$ at $s = 50$ by $m = 50$ different values and drop the observation $Z(s+1)$ in the model training, in order to get the prediction $\hat{Y}(s+1)$. By doing so, we are able to test the relationship and sensitivity of $Z(50)$ on $\hat{Y}(51)$ for both methods. The results in Figure 4(b) show that the Kriging predictor is linear in observation, while the DeepKriging provides an obviously nonlinear predictor in observation. This simulation study shows that although the prediction values of Kriging and DeepKriging are almost identical, the underlying relationship between predictor and observations is totally different.

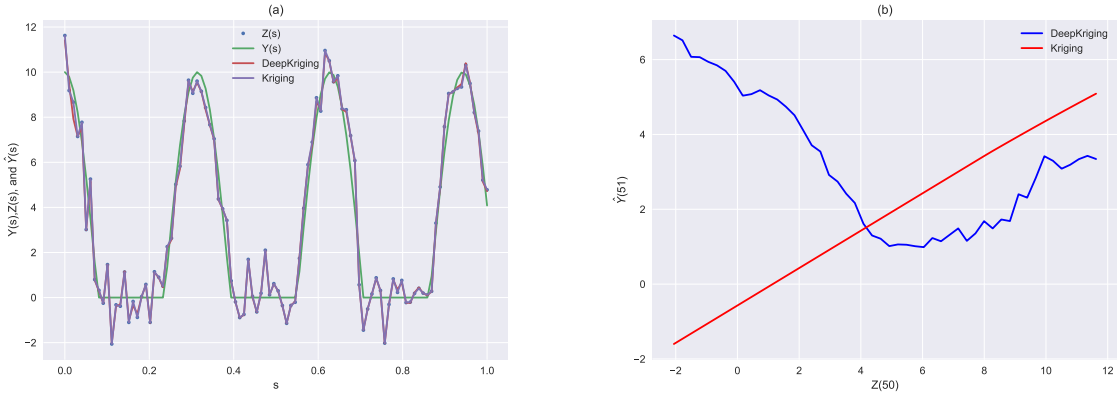

Figure 4: (a) The visualization of the simulated data and prediction. The observations (blue dots) Z(s) are generated by the the signal (green line) $Y(s)\mathbb{1}_{\{Y(s)>0\}} + \varepsilon(s)$, where $Y(s) = 10\cos(20s)$, with standard Gaussian noise. The prediction of Kriging (purple line) and DeepKriging (red line) are almost identical. (b) The relationship and sensitivity of $Z(50)$ on $\hat{Y}(51)$ from Kriging (red line) and DeepKriging (blue line) prediction. The x-axis shows 50 different values of $Z(50)$ in the model training and the y-axis shows 50 predicted values of $Y(51)$ provided that $Z(51)$ is not used.

5.4 Computational time of DeepKriging

Based on the same simulation setting in Section 5.3, we investigate the computational time of DeepKriging compared to Kriging with different sample size N. Both the number of epochs for DeepKriging and maximum number of iterations for Kriging are set to 200. Figure 5 shows the comparison results. Although Kriging is faster for small sample sizes ($N < 1,500$), DeepKriging is much more scalable when the sample size increases. For example, when $N = 12,800$, which is

the largest sample size we have considered, it takes more than 1.5 hours (5,663 seconds) to train a Kriging model, which makes Kriging computational infeasible for larger N. However, for the same data size, DeepKriging only costs 3.5 minutes (214 seconds) without GPU acceleration and 1.5 (94 seconds) minutes with a Tesla P100 GPU. Note that the GPU we use is freely accessed in Kaggle. For a more powerful GPU, the computational cost will be further reduced.

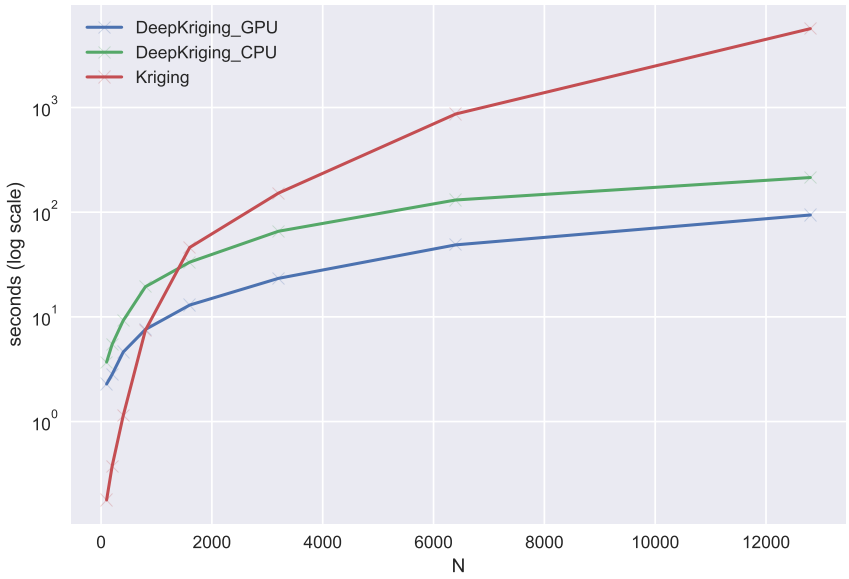

Figure 5: The computational time in seconds ($\log_{10}$ scale) of DeepKriging and Kriging for different number of observations N. The green line is the computational time of Kriging. The blue and orange lines are the computational time of DeepKriging based on a Tesla P100 GPU and a 2.5 GHz Intel Core i7 CPU, respectively.

6 Application to $PM_{2.5}$ exposure across the continental U.S.

$PM_{2.5}$, fine particulate matter of less than 2.5 μm, is a harmful air pollutant. Its adverse effects are associated with many diseases such as respiratory disease (Peng et al., 2009) and myocardial infarction (Peters et al., 2001) and thoroughly reviewed by Organization et al. (2013). Therefore, it is essential to obtain a high-resolution map of $PM_{2.5}$ exposure in order to investigate and assess its impact.

The Air Quality System (AQS) database provides ambient air monitoring data collected by U.S. Environmental Protection Agency (EPA) and other air pollution control agencies. The measurements of pollution concentrations from monitoring networks are the best characterization of the concentration of $PM_{2.5}$ at a given time and location. However, there are less than a thousand monitoring locations sparsely distributed over the U.S. so that the monitoring data are out of spatial and temporal alignment with health outcomes. On the other hand, it is known that $PM_{2.5}$ con-

centration is associated with meteorological conditions such as temperature and relative humidity (Jacob and Winner, 2009), where the meteorological data or data products are often easy to access have good spatial coverage and resolution. Therefore, the interpolation of $PM_{2.5}$ concentration by making use of from monitoring networks and other meteorological data has been a common and promising field research(Di et al., 2016).

The modeling and prediction of $PM_{2.5}$ concentration is challenging. First, $PM_{2.5}$ concentration data are obviously non-Gaussian, and thus the classical Kriging methods have some shortcomings. Second, the data from monitoring locations are irregular and sparse, but many interpolation methods require lattice data. Third, it is more important to understand the risk of high pollution and predict pollution levels, such as low, medium and high. Statistically, these two questions are related to estimating the probability over a threshold and classification problem, respectively. Quantile regression and convolutional neural networks have been employed recently to overcome some of the above issues (Reich et al., 2011b; Porter et al., 2015; Di et al., 2016). However, a method for all of the aforementioned tasks has not yet been sufficiently developed.

Therefore, we use the proposed DeepKriging method for spatial prediction of PM 2.5 concentrations using meteorological variables. Meteorological data were obtained from NCEP North American Regional Reanalysis (NARR) data, with a spatial resolution of about 32×32 km. A total of six meteorological variables are used in this study: 1) air temperature at 2 m, 2) relative humidity at 2 m, 3) accumulated total precipitation, 4) surface pressure, 5) u-component of wind, and 6) v-component of wind at 10 m. Since the units of the variables are different, we use min-max normalization to rescale the data.

As an example, we use the daily averaged data on June 05, 2019 of $PM_{2.5}$ concentration ($\mu g/m^3$) from 841 stations and six meteorological variables observed from $7,706$ locations. Since the coordinates from NARR and station data are not identical and some of stations are too close to each other, we match the data at the spatial resolution of NARR by averaging the PM measurements from the monitoring stations. After this matching, 604 stations remain available for the model training. We use the 10-fold cross-validation to verify the performance of DeepKriging. For comparison purposes, we also show the results from Kriging and the baseline DNN only with the six covariates and coordinates s. We calculate the mean squared error and mean absolute error, which are shown in the first two rows of Table 3.

Since DeepKriging is suitable for spatial data classification, we threshold the $PM_{2.5}$ concentrations by $12.0\mu g/m^3$, which is the standard for the annual mean and the threshold between "good" and "moderate" levels for the daily mean of EPA national ambient air quality standards (NAAQS) (EPA, 2012). We do not use the daily standard because too few observations are above the standard. Based on the classified data, we can implement a binary classification by assuming the actual values of $PM_{2.5}$ concentration to be unknown. Since Kriging is not feasible for binary data prediction, we predict the continuous $PM_{2.5}$ concentrations and then threshold the Kriging predictions. We use 10-fold cross-validation to show the performance in the last row of Table 3. It can be seen that DeepKriging significantly outperforms the Kriging and baseline DNN in terms of both MSE and classification accuracy.

Table 3: The model performance based on the 10-fold cross-validation. We choose the mean squared error (MSE), mean absolute error (MAE), and the classification accuracy (ACC) for predicting observation above $12.0\mu g/m^3$ as the validation criteria. The results are the mean and standard deviation (SD) of the 10 sets of validation errors.

Parameters	DeepKriging		Baseline DNN		Kriging	
	Mean	SD	Mean	SD	Mean	SD
MSE	**1.632**	.572	3.632	.925	3.361	.773
MAE	**.892**	.103	1.448	.162	1.365	.178
ACC	**95.2%**	2.6%	89.6%	4.8%	88.5%	4.6%

Once our model is fitted, we predict the $PM_{2.5}$ concentration, the level of pollution and the risk of high pollution level with the threshold 12 $\mu g/m^3$ at the unobserved locations using the NARR data. Figure 6(a) is the raw $PM_{2.5}$ station data from the AQS database. Figure 6(b) shows a smooth map of predicted $PM_{2.5}$ concentration from DeepKriging. The predicted distribution information can be used to obtain the predicted risk (shown in Figure 6(c)), defined as $\mathbb{P}\{PM_{2.5} > 12\ \mu g/m^3\}$, where 12 $\mu g/m^3$ is a breakpoint of the AQI category from EPA. To further visualize the difference between DeepKriging and Kriging, we also show the results with the Kriging prediction in Figure 6(d). The results show that DeepKriging provides more local features/patterns than Kriging.

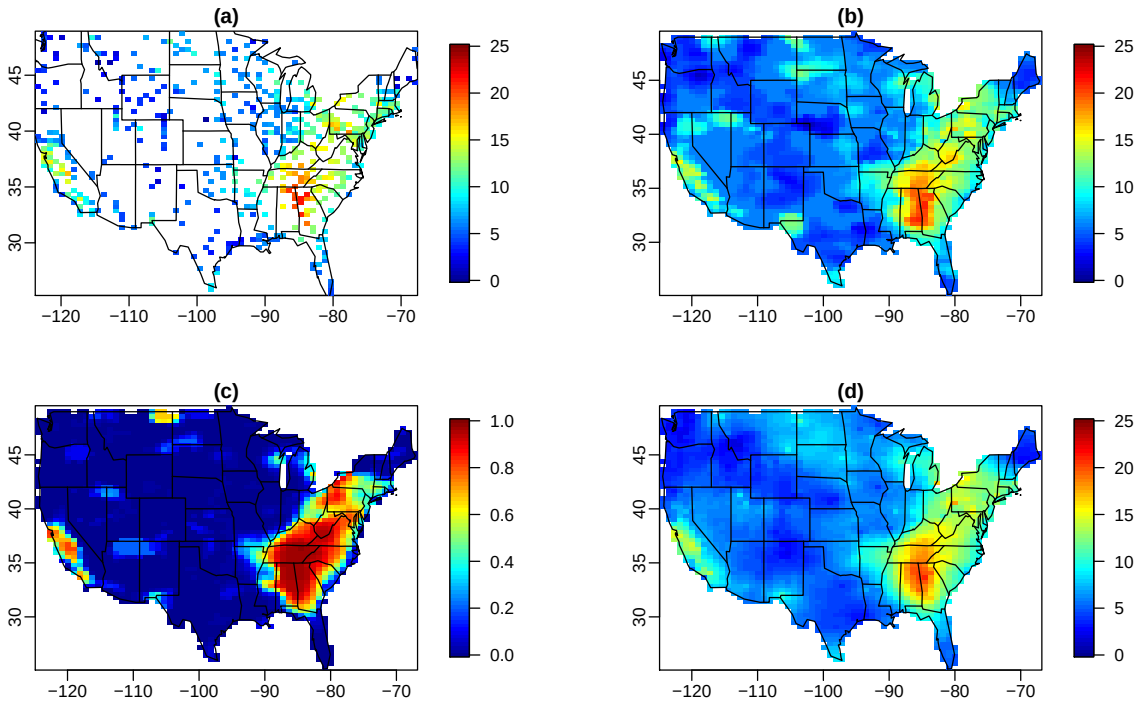

Figure 6: (a) The $PM_{2.5}$ concentration ($\mu g/m^3$) collected from monitoring stations. (b) The predicted $PM_{2.5}$ concentration using DeepKriging at 0.3 grid cells. (c) The predicted risk, $\mathbb{P}\{PM_{2.5} > 12\ \mu g/m^3\}$ of high pollution. (d) The predicted $PM_{2.5}$ concentration using Kriging at 0.3 grid cells.

7 Discussion

In this work, we have proposed a new spatial prediction model using neural networks and incorporated the spatial dependence by a set of basis functions. Our method does not assume parametric forms of covariance functions or data distributions, and is generally compatible with non-stationarity, non-linear relationships, and non-Gaussian data.

Classical Kriging methods consider their prediction as a linear combination of observations, which impedes their interaction with several machine learning frameworks. Some evidence of the equivalence between Kriging and radial basis functions interpolation has been known since 1981 in Matheron (1981). However, without the modern machine learning tools, only a linear combination and a limited number of radial basis functions have been investigated, which are viewed as a less favorable choice to Kriging (Dubrule, 1983, 1984). This work has provided a new perspective on deep learning and a large number of basis functions. We have shown the proposed method is superior to Kriging both theoretically and numerically in our simulation and real application. More importantly, the proposed DeepKriging framework connects the regression-based prediction and spatial prediction so that many other machine learning algorithms can be applied.

From the practical perspective, spatial prediction resembles the super-resolution and image inpainting in computer vision, in which CNNs are the dominant tools. Although both DeepKriging and CNNs can incorporate the spatial information, their objectives and applicable fields are different. CNNs are designed for images with gridded observed pixels as features and large training labels as the ground truth. They fill the missing pixels by filtering a patch of observed images and do not require other values as covariates. However, DeepKriging is designed for spatial fields or images only with sparse observations, where each observed pixel is viewed as a sample of response. It fills the missing pixels by the embedding of their coordinates and other covariates in DNN. Thus, DeepKriging is more suitable for environmental applications, for which high-resolution or full maps are often unavailable.

SUPPLEMENTARY MATERIAL

The supplementary materials contain more details of DeepKriging methods including the distribution prediction and uncertainty quantification (Section S1), computational details (Section S2), and the detailed network structure (Section S3). We also provide the source codes and data for reproducible research (Section S4).

8 Appendix

8.1 Proof of Lemma 1

Let the covariance matrix associated with the random process $\nu(\mathbf{s})$ be $\mathbf{\Sigma}^{\nu}$, where $\mathbf{\Sigma}^{\nu}_{i,j} = C(\mathbf{s}_i, \mathbf{s}_j)$. We can build a spatial random effect process $\mu(\mathbf{s}) = \boldsymbol{\phi}(\mathbf{s})^T\boldsymbol{\eta}$ with $\mathbb{C}\text{ov}(\boldsymbol{\eta}) = \mathbf{\Phi}_R^{-1}\mathbf{\Sigma}^{\nu}(\mathbf{\Phi}_R^{-1})^T$, where $\mathbf{\Phi} = \{\boldsymbol{\phi}(\mathbf{s}_1), \ldots, \boldsymbol{\phi}(\mathbf{s}_N)\}$ is an $N \times K$ basis matrix with rank N, and $\mathbf{\Phi}_R^{-1}$ is the right inverse of $\mathbf{\Phi}$.

From the result of Banerjee (1973), the right inverse exists. Hence, $\mathbb{C}\text{ov}(\boldsymbol{\eta})$ is also a valid covariance matrix and $\mathbb{C}\text{ov}(\boldsymbol{\mu}) = \mathbb{C}\text{ov}(\boldsymbol{\nu}) = \boldsymbol{\Sigma}^{\nu}$, where $\boldsymbol{\mu} = \{\mu(\mathbf{s}_1), \ldots, \mu(\mathbf{s}_N)\}^T$ and $\boldsymbol{\nu} = \{\nu(\mathbf{s}_1), \ldots, \nu(\mathbf{s}_N)\}^T$.

Therefore, the Kriging prediction of $\mu(\mathbf{s})$ is the same as that of $\nu(\mathbf{s})$ as shown in the Equation (2), such that $\widehat{Y}_{\mu}^{\text{UK}}(\mathbf{s}_0) = \widehat{Y}_{\nu}^{\text{UK}}(\mathbf{s}_0)$. On the other hand, based on the spatial random effect representation of $\mu(\mathbf{s})$, we can get the equivalent fixed rank Kriging prediction shown in the Equation (6), such that $\widehat{Y}_{\mu}^{\text{UK}}(\mathbf{s}_0) = \widehat{Y}_{\mu}^{\text{FRK}}(\mathbf{s}_0) = \mathbf{x}(\mathbf{s}_0)^T\widehat{\boldsymbol{\beta}} + \boldsymbol{\phi}(\mathbf{s}_0)^T\widehat{\boldsymbol{\alpha}}$. Therefore, $\widehat{Y}_{\nu}^{\text{UK}}(\mathbf{s}_0) = \mathbf{x}(\mathbf{s}_0)^T\widehat{\boldsymbol{\beta}} + \boldsymbol{\phi}(\mathbf{s}_0)^T\widehat{\boldsymbol{\alpha}}$. □

8.2 Proof of Lemma 2

Without loss of generality, we consider the mean squared loss in the prediction. The universal approximation theorem (Theorem 2.3.1 of Csáji (2001)) shows that $\forall f \in \mathbb{C}(\mathcal{X}(\mathbf{s}_0))$, $\forall \varepsilon > 0$: $\exists n \in \mathbb{N}$ and certain choice of weights and biases, such that $\|f(\mathbf{s}_0) - A_n f(\mathbf{s}_0)\| < \varepsilon$, where $A_n f$ is the mapping from the standard multilayer feed-forward networks with a single hidden layer that contains n hidden neurons. Then we have

$$\lim_{n\to\infty} \|f(\mathbf{s}_0) - A_n^{\text{opt}} f(\mathbf{s}_0)\|^2 = 0, \tag{8}$$

where $A_n^{\text{opt}} f(\mathbf{s}_0)$ is the optimal neural network that approximating $f(\mathbf{s}_0)$.

In the context of spatial prediction, suppose the optimal prediction in $\mathbb{C}(\mathcal{X}(\mathbf{s}_0))$ under, $\widehat{Y}_{\mathbb{C}(\mathcal{X}_\phi)}^{\text{opt}}(\mathbf{s}_0)$, minimize the mean squared loss, i.e., $\widehat{Y}_{\mathbb{C}(\mathcal{X}_\phi)}^{\text{opt}}(\mathbf{s}_0) = \underset{\widehat{Y}(\mathbf{s}_0)\in\mathbb{C}(\mathcal{X}(\mathbf{s}_0))}{\text{argmin}} \mathbb{E}\{\|\widehat{Y}(\mathbf{s}_0) - Y(\mathbf{s}_0)\|^2\}$. Then, the mean squared loss of $\widehat{Y}_{\mathcal{F}_{DK}}^{\text{opt}}(\mathbf{s}_0)$ satisfies

$$0 \leq \mathbb{E}\{\|\widehat{Y}_{\mathcal{F}_{DK}}^{\text{opt}}(\mathbf{s}_0) - Y(\mathbf{s}_0)\|^2\} - \mathbb{E}\{\|\widehat{Y}_{\mathbb{C}(\mathcal{X}_\phi)}^{\text{opt}}(\mathbf{s}_0) - Y(\mathbf{s}_0)\|^2\} \leq \mathbb{E}\{\|\widehat{Y}_{\mathcal{F}_{DK}}^{\text{opt}}(\mathbf{s}_0) - \widehat{Y}_{\mathbb{C}(\mathcal{X}_\phi)}^{\text{opt}}(\mathbf{s}_0)\|^2\}$$

Let $f(\mathbf{s}_0) = \widehat{Y}_{\mathbb{C}(\mathcal{X}_\phi)}^{\text{opt}}(\mathbf{s}_0) \in \mathbb{C}(\mathcal{X}(\mathbf{s}_0))$ in (8), then $A_n^{\text{opt}} f(\mathbf{s}_0) = \widehat{Y}_{\mathcal{F}_{DK}}^{\text{opt}}(\mathbf{s}_0)$ based on the definition, hence we have $\lim_{n\to\infty} \|\widehat{Y}_{\mathcal{F}_{DK}}^{\text{opt}}(\mathbf{s}_0) - \widehat{Y}_{\mathbb{C}(\mathcal{X}_\phi)}^{\text{opt}}(\mathbf{s}_0)\|^2 = 0$. Also note that

$$\|\widehat{Y}_{\mathcal{F}_{DK}}^{\text{opt}}(\mathbf{s}_0) - \widehat{Y}_{\mathbb{C}(\mathcal{X}_\phi)}^{\text{opt}}(\mathbf{s}_0))\|^2 \leq \|\widehat{Y}_{\mathbb{C}(\mathcal{X}_\phi)}^{\text{opt}}(\mathbf{s}_0) - Y(\mathbf{s}_0)\|^2 + \|\widehat{Y}_{\mathcal{F}_{DK}}^{\text{opt}}(\mathbf{s}_0) - Y(\mathbf{s}_0)\|^2 \leq 2\|\widehat{Y}_{\mathcal{F}_{DK}}^{\text{opt}}(\mathbf{s}_0) - Y(\mathbf{s}_0)\|^2$$

and $\mathbb{E}\{\|\widehat{Y}_{\mathcal{F}_{DK}}^{\text{opt}}(\mathbf{s}_0) - Y(\mathbf{s}_0)\|\}^2 < \infty$ by assumptions. Using the dominated convergence theorem, we have $\lim_{n\to\infty} \mathbb{E}\{\|\widehat{Y}_{\mathcal{F}_{DK}}^{\text{opt}}(\mathbf{s}_0) - \widehat{Y}_{\mathbb{C}(\mathcal{X}_\phi)}^{\text{opt}}(\mathbf{s}_0)\|^2\} = \mathbb{E}\{\lim_{n\to\infty} \|\widehat{Y}_{\mathcal{F}_{DK}}^{\text{opt}}(\mathbf{s}_0) - \widehat{Y}_{\mathbb{C}(\mathcal{X}_\phi)}^{\text{opt}}(\mathbf{s}_0)\|^2\} = 0$. As a result,

$$\widehat{Y}_{\mathcal{F}_{DK}}^{\text{opt}}(\mathbf{s}_0) = \widehat{Y}_{\mathbb{C}(\mathcal{X}_\phi)}^{\text{opt}}(\mathbf{s}_0) = \underset{Y(\mathbf{s})\in\mathbb{C}(\mathcal{X}_\phi)}{\text{argmin}} \mathbb{E}\{\|\widehat{Y}(\mathbf{s}_0) - Y(\mathbf{s}_0)\|^2\}\}. \quad \square$$

8.3 Proof of Theorem 2

Recall that the covariance function of $Y(\mathbf{s})$ is $C(\mathbf{s}_i, \mathbf{s}_j) = \mathbb{E}\{Y(\mathbf{s}_i)Y(\mathbf{s}_j)\} = \sigma_b^2 + \sigma_w^2\mathbb{E}\{a_{11}(\mathbf{s}_i)a_{11}(\mathbf{s}_j)\}$. Note that $\mathbb{E}\{a_{11}(\mathbf{s}_i)a_{11}(\mathbf{s}_j)\} = \frac{1}{2}[\mathbb{V}\text{ar}\{a_{11}(\mathbf{s}_i)\} + \mathbb{V}\text{ar}\{a_{11}(\mathbf{s}_j)\}] - \frac{1}{2}\mathbb{E}\{[a_{11}(\mathbf{s}_i) - a_{11}(\mathbf{s}_j)]^2\}$.

Therefore, when $\mathbf{s}_i$ is close to $\mathbf{s}_j$, we have $\psi_1(x) - \psi_1(y) = \alpha(x - y)$ for a smooth activation function, where α is a scaling coefficient. When the covariates $\mathbf{x}(\mathbf{s})$ is not available, we have $a_{11}(\mathbf{s}) = \psi_1(b_{11} + \sum_{k=1}^K W_{1k}\phi_k(\mathbf{s}))$. Thus $\mathbb{E}\{[a_{11}(\mathbf{s}_i) - a_{11}(\mathbf{s}_j)]^2\} = (\alpha\sigma_w)^2 \sum_{k=1}^K \{\phi_k(\mathbf{s}_i) - \phi_k(\mathbf{s}_j)\}^2$

Then the covariance function for nearby locations is $C(\mathbf{s}_i, \mathbf{s}_j) = \sigma_b^2 + \frac{\sigma_w^2}{2}[\mathbb{V}\text{ar}\{a_{11}(\mathbf{s}_i)\} + \mathbb{V}\text{ar}\{a_{11}(\mathbf{s}_j)\}] - (\alpha\sigma_w^2)^2 \sum_{k=1}^K \{\phi_k(\mathbf{s}_i) - \phi_k(\mathbf{s}_j)\}^2 \equiv v(\mathbf{s}_i) + v(\mathbf{s}_j) - c\|\boldsymbol{\phi}(\mathbf{s}_i) - \boldsymbol{\phi}(\mathbf{s}_j)\|^2$ □

References

Adgate, J. L., Ramachandran, G., Pratt, G., Waller, L., and Sexton, K. (2002). Spatial and temporal variability in outdoor, indoor, and personal PM2.5 exposure. *Atmospheric Environment*, 36(20):3255–3265.

Adler, R. J. (2010). *The Geometry of Random Fields*. SIAM.

Anselin, L. (2001). Spatial econometrics. *A Companion to Theoretical Econometrics*, 310330.

Austin, M. (2002). Spatial prediction of species distribution: an interface between ecological theory and statistical modelling. *Ecological Modelling*, 157(2-3):101–118.

Ba, S., Joseph, V. R., et al. (2012). Composite gaussian process models for emulating expensive functions. *The Annals of Applied Statistics*, 6(4):1838–1860.

Banerjee, K. S. (1973). Generalized inverse of matrices and its applications. *Technometrics*, 15(1):197–197.

Banerjee, S., Finley, A. O., Waldmann, P., and Ericsson, T. (2010). Hierarchical spatial process models for multiple traits in large genetic trials. *Journal of the American Statistical Association*, 105(490):506–521.

Banerjee, S., Gelfand, A. E., Finley, A. O., and Sang, H. (2008). Gaussian predictive process models for large spatial data sets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(4):825–848.

Carr, J. R. and Mao, N.-h. (1993). A general form of probability kriging for estimation of the indicator and uniform transforms. *Mathematical Geology*, 25(4):425–438.

Cho, Y. and Saul, L. K. (2009). Kernel methods for deep learning. In *Advances in neural information processing systems*, pages 342–350.

Cracknell, M. J. and Reading, A. M. (2014). Geological mapping using remote sensing data: A comparison of five machine learning algorithms, their response to variations in the spatial distribution of training data and the use of explicit spatial information. *Computers & Geosciences*, 63:22–33.

Cressie, N. (1990). The origins of kriging. *Mathematical Geology*, 22(3):239–252.

Cressie, N. (2015). *Statistics for Spatial Data*. John Wiley & Sons.

Cressie, N. and Johannesson, G. (2008). Fixed rank kriging for very large spatial data sets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(1):209–226.

Csáji, B. C. (2001). Approximation with artificial neural networks. *Faculty of Sciences, Etvs Lornd University, Hungary*, 24:48.

DeGroot, M. H. (2005). *Optimal Statistical Decisions*, volume 82. John Wiley & Sons.

Di, Q., Kloog, I., Koutrakis, P., Lyapustin, A., Wang, Y., and Schwartz, J. (2016). Assessing PM2.5 exposures with high spatiotemporal resolution across the continental United States. *Environmental Science & Technology*, 50(9):4712–4721.

Diggle, P. J., Thomson, M. C., Christensen, O., Rowlingson, B., Obsomer, V., Gardon, J., Wanji, S., Takougang, I., Enyong, P., Kamgno, J., et al. (2007). Spatial modelling and the prediction of loa loa risk: decision making under uncertainty. *Annals of Tropical Medicine & Parasitology*, 101(6):499–509.

Dubrule, O. (1983). Two methods with different objectives: splines and kriging. *Journal of the International Association for Mathematical Geology*, 15(2):245–257.

Dubrule, O. (1984). Comparing splines and kriging. *Computers & Geosciences*, 10(2-3):327–338.

EPA, U. (2012). National ambient air quality standards (naaqs). https://www.epa.gov/criteria-air-pollutants/naaqs-table. Date accessed: [Dec 15, 2019].

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360.

Fan, J., Ma, C., and Zhong, Y. (2019). A selective overview of deep learning. *arXiv preprint arXiv:1904.05526*.

Franchi, G., Yao, A., and Kolb, A. (2018). Supervised deep kriging for single-image super-resolution. In *German Conference on Pattern Recognition*, pages 638–649. Springer.

Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The Elements of Statistical Learning*. Springer.

Fuentes, M. (2002). Spectral methods for nonstationary spatial processes. *Biometrika*, 89(1):197–210.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press.

GPy (since 2012). GPy: A Gaussian process framework in python. http://github.com/SheffieldML/GPy.

Gulli, A. and Pal, S. (2017). *Deep Learning with Keras*. Packt Publishing Ltd.

Heaton, M. J., Datta, A., Finley, A. O., Furrer, R., Guinness, J., Guhaniyogi, R., Gerber, F., Gramacy, R. B., Hammerling, D., Katzfuss, M., et al. (2019). A case study competition among methods for analyzing large spatial data. *Journal of Agricultural, Biological and Environmental Statistics*, 24(3):398–425.

Hennessey Jr, J. P. (1977). Some aspects of wind power statistics. *Journal of Applied Meteorology*, 16(2):119–128.

Higdon, D. (2002). Space and space-time modeling using process convolutions. In *Quantitative methods for current environmental issues*, pages 37–56. Springer.

Ingebrigtsen, R., Lindgren, F., and Steinsland, I. (2014). Spatial models with explanatory variables in the dependence structure. *Spatial Statistics*, 8:20–38.

Jacob, D. J. and Winner, D. A. (2009). Effect of climate change on air quality. *Atmospheric Environment*, 43(1):51–63.

Journel, A. G. (1983). Nonparametric estimation of spatial distributions. *Journal of the International Association for Mathematical Geology*, 15(3):445–468.

Katzfuss, M. (2017). A multi-resolution approximation for massive spatial datasets. *Journal of the American Statistical Association*, 112(517):201–214.

Ketkar, N. et al. (2017). *Deep Learning with Python*. Springer.

Kleiber, W. and Nychka, D. W. (2015). Equivalent kriging. *Spatial Statistics*, 12:31–49.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

Lee, J., Bahri, Y., Novak, R., Schoenholz, S. S., Pennington, J., and Sohl-Dickstein, J. (2017). Deep neural networks as gaussian processes. *arXiv preprint arXiv:1711.00165*.

Lehmann, E. L. and Casella, G. (2006). *Theory of point estimation*. Springer Science & Business Media.

Li, R., Bondell, H. D., and Reich, B. J. (2019). Deep distribution regression. *arXiv preprint arXiv:1903.06023*.

Li, Y. and Sun, Y. (2019). Efficient estimation of non-stationary spatial covariance functions with application to high-resolution climate model emulation. *Statistica Sinica*, 29(3):1209–1231.

Matheron, G. (1963). Principles of geostatistics. *Economic Geology*, 58(8):1246–1266.

Matheron, G. (1976). A simple substitute for conditional expectation: the disjunctive kriging. In *Advanced geostatistics in the mining industry*, pages 221–236. Springer.

Matheron, G. (1981). Splines and kriging: their formal equivalence. *Down-to-Earth-Statistics: Solutions Looking for Geological Problems*, pages 77–95.

Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., and Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. *Journal of Big Data*, 2(1):1.

Neal, R. M. (2012). *Bayesian Learning for Neural Networks*, volume 118. Springer Science & Business Media.

Nychka, D., Bandyopadhyay, S., Hammerling, D., Lindgren, F., and Sain, S. (2015). A multiresolution gaussian process model for the analysis of large spatial datasets. *Journal of Computational and Graphical Statistics*, 24(2):579–599.

Organization, W. H. et al. (2013). Health effects of particulate matter. *Policy implications for countries in eastern Europe, Caucasus and central Asia*, 1(1):2–10.

Paciorek, C. J. and Schervish, M. J. (2004). Nonstationary covariance functions for Gaussian process regression. In *Advances in Neural Information Processing Systems*, pages 273–280.

Peng, R. D., Bell, M. L., Geyh, A. S., McDermott, A., Zeger, S. L., Samet, J. M., and Dominici, F. (2009). Emergency admissions for cardiovascular and respiratory diseases and the chemical composition of fine particle air pollution. *Environmental Health Perspectives*, 117(6):957–963.

Peters, A., Dockery, D. W., Muller, J. E., and Mittleman, M. A. (2001). Increased particulate air pollution and the triggering of myocardial infarction. *Circulation*, 103(23):2810–2815.

Porter, W. C., Heald, C. L., Cooley, D., and Russell, B. (2015). Investigating the observed sensitivities of air-quality extremes to meteorological drivers via quantile regression. *Atmospheric Chemistry and Physics*, 15(18):10349–10366.

Posch, K., Steinbrener, J., and Pilz, J. (2019). Variational inference to measure model uncertainty in deep neural networks. *arXiv preprint arXiv:1902.10189*.

Reich, B. J., Eidsvik, J., Guindani, M., Nail, A. J., and Schmidt, A. M. (2011a). A class of covariate-dependent spatiotemporal covariance functions. *The annals of applied statistics*, 5(4):2265.

Reich, B. J., Fuentes, M., and Dunson, D. B. (2011b). Bayesian spatial quantile regression. *Journal of the American Statistical Association*, 106(493):6–20.

Rimstad, K. and Omre, H. (2014). Skew-Gaussian random fields. *Spatial Statistics*, 10:43–62.

Sampson, P. D., Richards, M., Szpiro, A. A., Bergen, S., Sheppard, L., Larson, T. V., and Kaufman, J. D. (2013). A regionalized national universal Kriging model using partial least squares regression for estimating annual PM2.5 concentrations in epidemiology. *Atmospheric Environment*, 75:383–392.

Stein, M. L. (2014). Limitations on low rank approximations for covariance matrices of spatial data. *Spatial Statistics*, 8:1–19.

Tobler, W. R. (1970). A computer movie simulating urban growth in the detroit region. *Economic geography*, 46(sup1):234–240.

Vidakovic, B. (2009). *Statistical Modeling by Wavelets*, volume 503. John Wiley & Sons.

Wahba, G. (1990). *Spline Models for Observational Data*, volume 59. SIAM.

Waller, L. A. and Gotway, C. A. (2004). *Applied Spatial Statistics for Public Health Data*, volume 368. John Wiley & Sons.

Wang, H., Guan, Y., and Reich, B. J. (2019). Nearest-neighbor neural networks for geostatistics. *arXiv preprint arXiv:1903.12125*.

Whittle, P. (1954). On stationary processes in the plane. *Biometrika*, pages 434–449.

Xiong, Y., Chen, W., Apley, D., and Ding, X. (2007). A non-stationary covariance-based kriging method for metamodelling in engineering design. *International Journal for Numerical Methods in Engineering*, 71(6):733–756.

Xu, G. and Genton, M. G. (2017). Tukey g-and-h random fields. *Journal of the American Statistical Association*, 112(519):1236–1249.

Zhao, H., Gallo, O., Frosio, I., and Kautz, J. (2015). Loss functions for neural networks for image processing. *arXiv preprint arXiv:1511.08861*.

DeepKriging: Spatially Dependent Deep Neural Networks for Spatial Prediction

Yuxiao Li[1], Ying Sun[1], Brian J Reich [2]

July 28, 2020

SUPPLEMENTARY MATERIAL

The supplementary materials contain more details of DeepKriging methods including the distribution prediction and uncertainty quantification (Section S1), computational details (Section S2), and the detailed network structure (Section S3). We also provide the source codes and data for reproducible research (Section S4).

[1] Statistics Program, King Abdullah University of Science and Technology, Thuwal 23955-6900, Saudi Arabia. E-mail: yuxiao.li@kaust.edu.sa; ying.sun@kaust.edu.sa

[2] North Carolina State University, Department of Statistics, Campus Box 8203, 5212 SAS Hall, Raleigh, NC 27695. Email: brian_reich@ncsu.edu

This research is supported by King Abdullah University of Science and Technology (KAUST), Office of Sponsored Research (OSR) under Award No: OSR-2019-CRG7- 3800.

S1 Distribution prediction and uncertainty quantification of DeepKriging

Classical deep learning models typically do not provide uncertainty quantification. To tackle this challenge, we propose to quantify the uncertainty of the DeepKriging prediction by the conditional distribution of $Y(\mathbf{s}_0)|\mathbf{Z}, \mathbf{x}_\phi(\mathbf{s}_0)$ at an unobserved location $\mathbf{s}_0$. We denote the probability density function (PDF) as $f^{\text{UQ}}(y|x)$. Unlike in the Kriging, the conditional distribution is assumed to be Gaussian, and we estimate the entire conditional distribution non-parametrically.

Several deep learning methods are available to estimate the density function. The Bayesian framework (Gal and Ghahramani, 2016; Posch et al., 2019) is the most commonly used for this task. It usually assumes a certain prior distribution of each parameter, and then obtains the density by the posterior predictive distribution. Mixture density networks (Bishop, 1994) are another tool that assume a parametric form of the density and estimates the unknown parameters by neural networks. However, both of the methods require an assumption of data distribution. Recently, Li et al. (2019) proposed a completely distribution-free method for density estimation called deep distribution regression (DDR), where the density was approximated using histograms with discrete bins, and the discretized density was estimated by multi-class classification. Although the method was designed for regression, we extend the idea to the framework of DeepKirging for predictive distribution estimation and call the method deep distribution spatial prediction (DDSP).

Specifically, we denote the support of $f^{\text{UQ}}(y|x)$ by $[l, u]$, where l and u are the proposed lower and upper bound of domain of the density. The interval can be partitioned into $M+1$ bins, $T_m = [c_{m-1}, c_m)$, by M cut-points such that $c_0 < c_1 < \cdots < c_M < c_{M+1}$, where $c_0 = l$ and $c_{M+1} = u$. Let $|T_m|$ be the length of the m-th bin and $p_m(\mathbf{s}_0) = \mathbb{P}\{Y(\mathbf{s}_0) \in T_m|\mathbf{Z}, \mathbf{x}_\phi(\mathbf{s}_0)\}$ be the conditional probability that $Y(\mathbf{s}_0)$ falls into the m-th bin. Then $f^{\text{UQ}}(y|x)$ is approximated by $M+1$ constant functions, $p_m(\mathbf{s}_0)/|T_m|, m = 1, \ldots, M+1$.

The density prediction is specified as:

$$\hat{f}^{\text{UQ}}(y|x) = \sum_{m=1}^{M+1} \frac{\widehat{p}_m(\mathbf{s}_0)}{|T_m|} \mathbb{1}\{y \in T_m\}. \tag{S1.1}$$

The crucial step in Equation (S1.1) is to estimate the bin probability $\{\widehat{p}_1(\mathbf{s}_0), \ldots, \widehat{p}_{M+1}(\mathbf{s}_0)\}$ using a classification model, so that neural networks can be applied. Li et al. (2019) suggested different ways for loss functions and bin partitioning. The most natural way is to use multi-class classification with fixed bins. In the output layer, a softmax function is applied to ensure that $\{\widehat{p}_1(\mathbf{s}_0), \ldots, \widehat{p}_{M+1}(\mathbf{s}_0)\}$ constitutes a valid probability vector. The loss function for this case is the negative multinomial log-likelihood function, which is equivalent to the multi-class cross entropy loss: $\sum_{n=1}^{N} \sum_{m=1}^{M+1} \mathbb{1}\{Z(\mathbf{s}_n) \in T_m\} \log\{p_m(\mathbf{s}_n)\}$.

As an extension of Wasserman (2006) regarding the consistency of histogram, Li et al. (2019) shows that in the logistic regression case, the density predictors from DDR with multi-class loss function and equally spaced fixed bins is consistent. That is, $\int_l^u [\hat{f}^{\text{UQ}}(y|x) - f^{\text{UQ}}(y|x)]^2 dy \xrightarrow{N\to\infty} 0$.

In practice, however, the estimation of a continuous density function by the multi-class fixed

classification is sensitive to the choice of bins. Therefore, in the DeepKriging, we use the improved option with joint binary cross entropy loss function (JBCE) and ensembles. The JBCE function is specified as

$$JBCE = \sum_{n=1}^{N}\sum_{m=1}^{M}\left[\mathbb{1}\{Z(\mathbf{s}_n) \le c_m\}\log\{F(c_m;\mathbf{s}_n)\} + \mathbb{1}\{Z(\mathbf{s}_n) > c_m\}\log\{1-F(c_m;\mathbf{s}_n)\}\right], \quad \text{(S1.2)}$$

where $F(c_m;\mathbf{s}_n) = \sum_{i=1}^{m} p_i(\mathbf{s}_n)$ and $c_m, m = 1,\ldots,M$ are the M cut-points.

The DDSP may provide a non-smooth density. Thus, an ensemble method can be applied for further adjustment by fitting I independent classifications and computing the average of classifiers. Algorithm 1 provides the procedure of obtaining density prediction of DeepKriging with the ensemble of random partitioning:

Algorithm 1: The algorithm of density prediction of DeepKriging with ensemble random partitioning. We ensemble I independent classifications, for each of which M bins are chosen randomly.

for *i =1:I* **do**
- Draw M cut-points from Uniform(0,1);
- Sort the cut-points as $c_{i1},\ldots,c_{iM}$;
- Assign $Y(\mathbf{s}_0)$ to M bins, where $T_{im} = [c_{i,m-1}, c_{im})$;
- Train the classifier to estimate the probabilities $\widehat{p}_{i1}(\mathbf{s}_0),\ldots,\widehat{p}_{i,M+1}(\mathbf{s}_0)$;

end

Result: $\hat{f}^{\mathrm{UQ}}(y|x) = \sum_{i=1}^{I}\sum_{m=1}^{M+1}\frac{\widehat{p}_{im}(\mathbf{s}_0)}{|T_{im}|}\mathbb{1}\{y \in T_{im}\}$

Note that although our density regression provides an uncertainty quantification of the response variable without any distribution assumption, it may bring new uncertainties from the neural networks. Therefore, the GP process representation may be a better way for uncertainties if it is quantified by the standard deviation, even though the computational burden will increase.

In addition, the density prediction requires further computations if we apply ensemble, where each step in the ensemble will implement classification separately with a new choice of random cut-points. However, the main objective of the ensemble in the density prediction is to obtain a smooth density. If the research goal is to get uncertainties using quantiles without the density curve, then the ensemble is not always needed. Another hyperparameter we need to specify in the density prediction is the bin size $M+1$. Typically, large M will provide a more complex pattern of the density, and small M will give a more stable estimation. In practice, we can follow the Freedman–Diaconis rule (Freedman and Diaconis, 1981) as a robust estimation in the histogram approximation such that $M = \left\lfloor \{\max(\mathbf{Z}) - \min(\mathbf{Z})\}\frac{\sqrt[3]{N}}{2\times \mathrm{IQR}(\mathbf{Z})} \right\rfloor$, where $\mathrm{IQR}(\mathbf{Z})$ is the interquartile range of the observations.

S1.1 The uncertainty quantification of DeepKriging: a simulation study

The simulated data is generated by a Gaussian mixture model:

$$Z(s) = \{\sin(5s) + 0.7 + \tau_1(s)\}\pi(s) + \{2\sin(8s) + \tau_2(s))\}(1-\pi(s)),$$

where $\tau_1(s) \overset{iid}{\sim} N(0, 0.2^2)$, $\tau_2(s) \overset{iid}{\sim} N(0, 0.3^2)$, $\pi(s) \overset{iid}{\sim} Bernoulli(0.5)$, and $\tau_2(s)$, $\tau_2(s)$, and $\pi(s)$ are mutually independent. Observations are drawn from $2,500$ regularly spaced locations in $[0, 1]$, where $2,000$ are training samples and 500 are testing samples.

Figure 1 shows the prediction results on both training and testing samples. The advantage of DeepKriging is obvious for the uncertainty quantification. The process is heteroscedastic with an obviously large variance when s is larger than 0.8. The heteroscedasticity is well captured by the DeepKriging but missed by the Kriging. As a result, the prediction band of the Kriging is too narrow for large s and two large for small s.

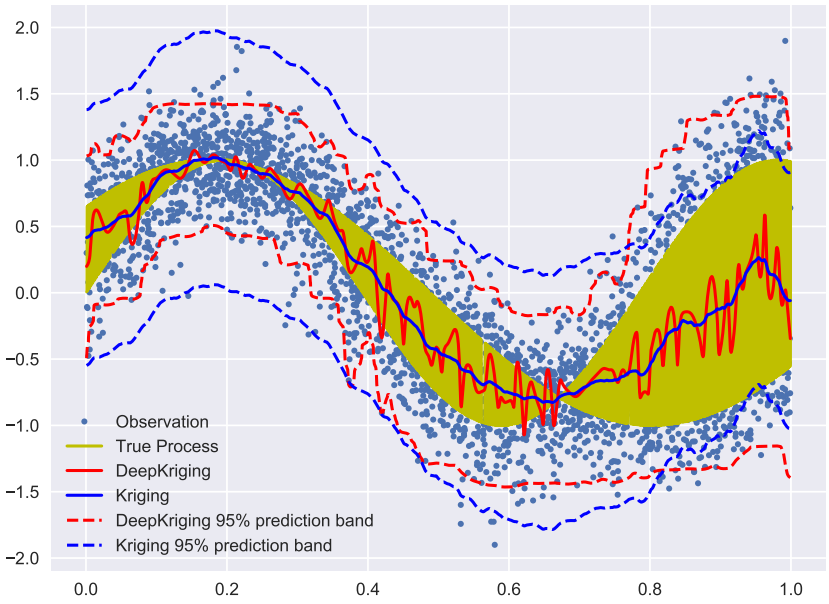

Figure 1: The prediction results of a Gaussian mixture model. The blue dots are the observations simulated from the Gaussian mixture model. The solid lines are the true process (yellow) and its prediction from the DeepKriging (red) and Kriging (blue) . The dashed lines are 95% prediction band from the DeepKriging (red) and Kriging (blue).

In addition, numerical evaluations on testing samples are used for the comparison based on the root mean squared error (RMSE) and average quantile loss (AQTL). AQTL is defined by $AQTL = \sum_{t=1}^{99} \sum_{n=1}^{N} (\{Z(s_n) - \widehat{Q}(t/100)\}[t/100 - \mathbb{1}\{Z(s_n) \leq \widehat{Q}(t/100)\}])$, where $\widehat{Q}(t/100)$ is the estimated t-th percentile. The results show that the RMSE of the Kriging (0.490) is comparable to DeepKriging (0.485). But, in terms of the quantile loss, the AQTL of the Kriging (60.47) is much larger than that in the DeepKriging (0.12), which further shows that Kriging is a only a good estimation in terms of the mean squared error including the mean and variance, but cannot reflect other important properties such as quantiles.

S2 Computational details of DeepKriging

The computation of DeepKriging depends on two aspects: choice of basis functions and network structure. We use the multi-resolution radial basis functions proposed in (Nychka et al., 2015), which provides a good choice of knots and bandwidth. They consider a multi-resolution compactly supported Wendland radial basis function (RBF) with the form $w(d) = \frac{(1-d)^6}{3}(35d^2+18d+3)\mathbb{1}\{0 \leq d \leq 1\}$, where $d = \|\mathbf{s}-\mathbf{u}\|/\theta$.

Here, we choose K_θ bandwidth and $K_{u\theta}$ knots for each bandwidth, indexed by k_θ and $k_{u\theta}$, then the number of RBFs is $K = \sum_{k_\theta=1}^{K_\theta} K_{u\theta}$ and each RBF is $\phi_{k_{u\theta},k_\theta}(\mathbf{s}) = w(d_{k_{u\theta},k_\theta}(\mathbf{s}))$, where $d_{k_{u\theta},k_\theta}(\mathbf{s}) = \|\mathbf{s}-\mathbf{u}_{k_{u\theta}}\|/\theta_{k_\theta}$. The scale factor or bandwidth, θ_{k_θ}, is set to be 2.5 times of the associated knots spacing. The level of resolution is determined by K_θ so that each θ_{k_θ} defines a lattice network of knots built in the spatial domain. In the k_θ-th level, K_u is chosen to be $K_u = (9 \times 2^{k_\theta-1}+1)^d$ to have non-overlapped knots with good coverage, where d is the spatial dimension. Nychka et al. (2015) proposes to use a four-level model so that there are $K = 10+19+37+73 = 139$ RBFs in one dimension and $K = 10^2+19^2+37^2+73^2 = 7159$ RBFs in two dimensions. As we have shown in Theorem 1, it is more favorable to have a large number of basis functions K. The choice of K is generally large enough, but for a massive dataset and to obtain $K \geq N$, we need $K_\theta = 1+\lceil \log_2(\sqrt[d]{N}/10)\rceil$ levels.

The time complexity of DeepKriging is about $O(N_{\text{neuron}})$, where N_{neuron} is the number of neurons. The computational cost of DeepKriging depends on the width and depth of the associated neural network. Recall that the time complexity of Kriging is $O(N^3)$. DeepKriging with fair complexity is more efficient than Kriging for large N. Moreover, the computation of neural networks is highly parallelizable and can be largely accelerated by GPU. The details of the acceleration are discussed in Bergstra et al. (2011) and implemented in Keras (Gulli and Pal, 2017), as a Python and R library in which our method is implemented. An illustration can be found in the supplementary materials.

S3 The network structure of DeepKriging

After the simulation on different types of datasets, we can summarize some of the findings of neural network structures. First, the dropout layer is the most useful regularization for DeepKriging when the overfitting occurs, typically when the data includes the outliers or the sample size is small. For a large dataset with almost a Gaussian distribution, dropout and other types of regularization are not very helpful. Batch-normalization is another useful strategy for DeepKriging since the covariates may have different units and the the scales of basis function may vary too much for irregularly spaced spatial data. Furthermore, we suggest to normalize the covariates first before we run the automated batch-normalization since the covariates and basis functions required different types of normalization. Last, if the data are irregularly spaced, the value of basis functions for some knots will be zero for all locations, since the basis function we choose is compactly supported and the knots may be far from any location. In this case, it is required to remove these knots and

related columns in the basis matrix. To summarize, the default setting of DeepKriging network is as follows:

1) Normalize (Min-max normalization) the observed covariates (features) $\mathbf{x}(\mathbf{s})$; 2) Build the embedding layer using a multi-resolution radial basis functions $\boldsymbol{\phi}(\mathbf{s})$ with the corresponding basis matrix $\boldsymbol{\Phi}$; 3) Remove the all-zero columns of the basis matrix $\boldsymbol{\Phi}$; 4) Add the first dense layer with 100 hidden neurons and ReLu activations; 5) Add the first dropout layer with 0.5 dropout rate; 6) Add the first batch-normalization layer; 7) Add the second dense layer with 100 hidden neurons and ReLu activations; 8) Add the second dropout layer with 0.5 dropout rate; 9) Add the third dense layer with 100 hidden neurons and ReLu activations; 10) Add the second batch-normalization layer; 11) Add the output layer.

S4 The source codes and data

- The source codes can be found in the following Github repository: https://github.com/aleksada/DeepKriging.
- The methods are implemented in Keras, a library of both Python and R. Most of the codes of our study are written in Python. However, we also include some simple examples of DeepKriging using Keras in R.
- The PM2.5 station data from AQS used in the application can be found in https://aqs.epa.gov/aqsweb/airdata/download_files.html.
- The meteorological data from NARR used in the application can be found in https://psl.noaa.gov/data/gridded/data.narr.html

References

Bergstra, J., Bastien, F., Breuleux, O., Lamblin, P., Pascanu, R., Delalleau, O., Desjardins, G., Warde-Farley, D., Goodfellow, I., Bergeron, A., et al. (2011). Theano: Deep learning on gpus with python. In *NIPS 2011, BigLearning Workshop, Granada, Spain*, volume 3, pages 1–48. Citeseer.

Bishop, C. M. (1994). Mixture density networks. *Technical Report*.

Freedman, D. and Diaconis, P. (1981). On the histogram as a density estimator: L2 theory. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 57(4):453–476.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059.

Gulli, A. and Pal, S. (2017). *Deep Learning with Keras*. Packt Publishing Ltd.

Li, R., Bondell, H. D., and Reich, B. J. (2019). Deep distribution regression. *arXiv preprint arXiv:1903.06023*.

Nychka, D., Bandyopadhyay, S., Hammerling, D., Lindgren, F., and Sain, S. (2015). A multiresolution gaussian process model for the analysis of large spatial datasets. *Journal of Computational and Graphical Statistics*, 24(2):579–599.

Posch, K., Steinbrener, J., and Pilz, J. (2019). Variational inference to measure model uncertainty in deep neural networks. *arXiv preprint arXiv:1902.10189*.

Wasserman, L. (2006). *All of Nonparametric Statistics*. Springer Science & Business Media.