Statistica Sinica, to appear

DeepKriging: Spatially Dependent Deep Neural Networks for Spatial Prediction

Wanfang Chen[1], Yuxiao Li[2], Brian J Reich[3] and Ying Sun[2]

[1] *East China Normal University*

[2] *King Abdullah University of Science and Technology*

[3] *North Carolina State University*

Abstract: In spatial statistics, a common objective is to predict values of a spatial process at unobserved locations by exploiting spatial dependence. 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 no longer optimal, and the associated variance is often overly optimistic. Although deep neural networks (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 DNN structure for spatial prediction, where the spatial dependence is captured by adding an embedding layer of spatial coordinates with basis functions. We show in theory and simulation studies that the proposed DeepKriging method has a direct link to Kriging in the Gaussian case, and it has multiple advantages over Kriging for non-Gaussian and non-stationary data, i.e., it provides non-linear predictions and thus has smaller approximation

errors, it does not require operations on covariance matrices and thus is scalable for large datasets, and with sufficiently many hidden neurons, it provides the optimal prediction in terms of model capacity. We further explore the possibility of quantifying prediction uncertainties based on density prediction without assuming any data distribution. Finally, we apply the method to predicting $PM_{2.5}$ concentrations across the continental United States.

Key words and phrases: Basis function, Deep learning, Feature embedding, Gaussian process, Spatial Prediction.

1. Introduction

Spatial prediction is at the heart of spatial and spatio-temporal statistics. It is aimed at predicting values of a spatial process at unobserved locations by accounting for the spatial dependence in the region of interest. Traditional applications of spatial prediction are in the fields of geological and environmental science (Cressie, 2015), and 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 the best linear unbiased prediction (BLUP), also referred to as Kriging (Matheron, 1963). Kriging prediction is a weighted average of observed data, where the weights are determined by the spatial covariance function or var-

iogram of the random process. Under the Gaussian assumption, Kriging also provides the full predictive distribution. Applying Kriging requires estimating the spatial covariance function, which is commonly assumed to be stationary. 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 the spatial covariance typically 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), but Kriging for more general spatial processes remains an open problem. Another drawback of Kriging is that it is computationally prohibitive for large spatial datasets, since it involves computing the inversion of an $N \times N$ covariance matrix, where N is the number of observed locations (Heaton et al., 2019), and the computation requires $O(N^3)$ time and $O(N^2)$ memory complexity based on the typical Cholesky decomposition approach.

Recently, deep learning or deep neural networks (DNNs) have become the most powerful prediction tools for a wide range of applications, espe-

cially in computer vision and natural language processing (LeCun et al., 2015). DNNs are effective for predictions with complex features such as non-linearity and non-stationarity, and they are computationally efficient in analyzing massive datasets using GPUs (Najafabadi et al., 2015). Therefore, it would be promising to apply DNNs to spatial predictions. However, classical DNNs cannot incorporate the spatial dependence appropriately. 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 claimed 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.

To tackle the above-mentioned problems, we develop an effective deep neural network for spatial prediction that

1) builds a direct link between DNNs and Kriging in spatial prediction;

2) models the spatial dependence through a set of basis functions;

3) does not require matrix operations and is scalable for large datasets;

4) provides a non-linear predictor in covariates or generally in observations;
5) has a Gaussian process representation, and brings more flexible spatial covariance structures than simply using the coordinates as features;
6) suits for different data types, e.g., non-Gaussian or non-stationary data;
7) potentially 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, similar to the original use of Kriging (Cressie, 1990), but by using deep neural networks. We also conduct simulation studies and apply our approach to the $PM_{2.5}$ concentration data across the continental United States to show the performance of DeepKriging compared to Kriging and other naive DNN methods. The rest of our paper is organized as follows. Section 2 introduces the construction of our DeepKriging method. Section 3 provides its theoretical properties. Section 4 presents some simulation studies to show the performance of DeepKriging. Section 5 applies DeepKriging to predict $PM_{2.5}$ concentration in the U.S. Section 6 summarizes the main results and suggests directions for future work.

2. Methodology

2.1 Deep learning in spatial prediction

Suppose $\mathbf{z} = \{z(\mathbf{s}_1), \ldots, z(\mathbf{s}_N)\}^T$ are measurements observed at N spatial locations from a real-valued spatial process $\{Y(\mathbf{s}) : \mathbf{s} \in D\},\ D \subseteq \mathbb{R}^d$. The goal of spatial prediction is to find the optimal predictor $\widehat{Y}^{\text{opt}}(\mathbf{s}_0)$ of the true process at an unobserved location $\mathbf{s}_0$, as a function of $\mathbf{z}$. In decision theory, $\widehat{Y}^{\text{opt}}(\mathbf{s}_0)$ is the minimizer of an expected loss function or risk function (DeGroot, 2005). That is,

$$\widehat{Y}^{\text{opt}}(\mathbf{s}_0) = \underset{\hat{Y}}{\text{argmin}}\ \mathbb{E}\{L(\widehat{Y}(\mathbf{s}_0), Y(\mathbf{s}_0))\} = \underset{\hat{Y}}{\text{argmin}}\ R(\widehat{Y}(\mathbf{s}_0), Y(\mathbf{s}_0)), \quad (2.1)$$

where $L(\cdot, \cdot)$ is a loss function and $R(\cdot, \cdot)$ is a risk function. Under the mean squared error (MSE) loss, the optimal predictor is $\widehat{Y}^{\text{opt}}(\mathbf{s}_0) = \mathbb{E}\{Y(\mathbf{s}_0)|\mathbf{z}\}$ if it is finite. This predictor has multiple good properties such as unbiasedness and asymptotic normality under regularity assumptions (Lehmann and Casella, 2006). In particular, if $Y(\mathbf{s}_0)$ and $\mathbf{z}$ are jointly Gaussian, the conditional mean is a linear combination of $\mathbf{z}$; if $Y(\mathbf{s}_0)$ and $\mathbf{z}$ are not jointly Gaussian, the conditional mean obtained with Gaussian assumption remains the best linear unbiased prediction (BLUP), which is called Kriging. However, as mentioned before, the Kriging predictor is sub-optimal for non-Gaussian data, and it is not scalable for large data size.

In this work, we use deep learning to approximate the optimal predictor $\widehat{Y}^{\text{opt}}(\mathbf{s}_0)$ in (2.1) by the output of the neural network. The optimal neural network predictor is given by $f_{\text{NN}}^{\text{opt}}(\mathbf{s}_0) = \text{argmin}_{f_{\text{NN}}} R\{f_{\text{NN}}(\mathbf{s}_0), Y(\mathbf{s}_0)\}$, where $f_{\text{NN}}(\cdot) \in \mathcal{F}$ can be any function in the function space $\mathcal{F}$ expressible by a family of neural networks, and $f_{\text{NN}}^{\text{opt}}(\cdot)$ is the best function in $\mathcal{F}$ in terms of minimizing a certain risk $R(\cdot, \cdot)$. The inputs of the neural network can be relevant covariates $\mathbf{x}(\mathbf{s}_0)$ and other features at $\mathbf{s}_0$. Typically, we write $f_{\text{NN}}(\mathbf{s}; \boldsymbol{\theta})$ as a parametric model with unknown parameters $\boldsymbol{\theta}$, which include the weights and biases in the neural network. Note that the optimal neural network predictor $f_{\text{NN}}^{\text{opt}}(\mathbf{s}_0)$ is practically unreachable since $Y(\mathbf{s}_0)$ is unknown. In practice, we approximate the predictor by minimizing the empirical loss function over the training set $\mathbf{z}$ (Goodfellow et al., 2016); i.e., the final predictor is $\widehat{Y}_{\text{NN}}(\mathbf{s}_0) = f_{\text{NN}}(\mathbf{s}_0; \widehat{\boldsymbol{\theta}})$, with

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\text{argmin}} \frac{1}{N} \sum_{n=1}^{N} L\{f_{\text{NN}}(\mathbf{s}_n; \boldsymbol{\theta}), z(\mathbf{s}_n)\}. \tag{2.2}$$

Applying this framework of classical neural network directly to spatial prediction is problematic in at least two aspects: classical DNNs does not account for the spatial dependence, and spatial prediction typically has limited observed features rather than excessive features in common applications of neural networks. In particular, assume that the spatial process $Y(\mathbf{s})$ is modeled by $Y(\mathbf{s}) = \mathbf{x}(\mathbf{s})^T\boldsymbol{\beta} + \nu(\mathbf{s})$, where $\mathbf{x}(\mathbf{s}) \in \mathbb{R}^P$ is a vec-

tor process of P known covariates, $\boldsymbol{\beta}$ is a vector of coefficients, and $\nu(\mathbf{s})$ is a spatially dependent and zero-mean random process with a generally non-stationary covariance function: $\text{Cov}(\nu(\mathbf{s}), \nu(\mathbf{s}')) = \text{C}(\mathbf{s}, \mathbf{s}')$. In neural networks, we usually assume that $Y(\mathbf{s})$ are mutually independent conditional on the features $\mathbf{x}(\mathbf{s})$. However, this assumption is not reasonable in spatial prediction because 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. Hence, 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 $\mathbf{s}$ (Cracknell and Reading, 2014). By doing that, the adjusted features become $\mathbf{x}^{adj}(\mathbf{s}) = (\mathbf{x}(\mathbf{s})^T, \mathbf{s})^T$. However, this does not help much in enlarging the feature space since usually the dimension of coordinates has $d \leq 3$. Moreover, the associated neural network may not be efficient, since if the true function is far from linear, it may take huge effort for the neural network to achieve a good approximation. For instance, the optimal predictor under the Gaussian assumption and MSE loss is the Kriging predictor, which is linear in $\mathbf{x}(\mathbf{s})$ but obviously non-linear in coordinates $\mathbf{s}$; this is a special

case where the natural structure of neural networks may not work.

Going deeper into the form of Kriging prediction may give us a hint about the appropriate way to incorporate the spatial dependence in the DNN. Suppose $\mathbf{z}$ is observed from a generalized additive model: $Z(\mathbf{s}) = Y(\mathbf{s}) + \varepsilon(\mathbf{s})$, where $Y(\mathbf{s}) = \mathbf{x}(\mathbf{s})^T\boldsymbol{\beta} + \nu(\mathbf{s})$ as defined above, and $\varepsilon(\mathbf{s})$ is a white noise process, called the nugget effect, with zero mean and variance $\sigma^2(\mathbf{s})$, caused by measurement inaccuracy and fine-scale variability. The (universal) Kriging prediction is

$$\widehat{Y}_{\mathrm{UK}}(\mathbf{s}_0) = \mathbf{x}(\mathbf{s}_0)^T\widehat{\boldsymbol{\beta}} + \mathbf{c}(\mathbf{s}_0)^T\boldsymbol{\Sigma}^{-1}(\mathbf{z} - \mathbf{X}\widehat{\boldsymbol{\beta}}), \tag{2.3}$$

where $\mathbf{X} = (\mathbf{x}(\mathbf{s}_1), \ldots, \mathbf{x}(\mathbf{s}_N))^T$ is an $N \times P$ matrix, $\mathbf{c}(\mathbf{s}_0) = \mathrm{Cov}(\mathbf{Z}, Z(\mathbf{s}_0))$, $\boldsymbol{\Sigma} = \mathrm{Cov}(\mathbf{Z}, \mathbf{Z}^T)$, and $\widehat{\boldsymbol{\beta}} = (\mathbf{X}^T\boldsymbol{\Sigma}^{-1}\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{\Sigma}^{-1}\mathbf{z}$. The spatial dependence is incorporated in $\widehat{Y}_{\mathrm{UK}}(\mathbf{s}_0)$ via a linear function of the covariance vector $\mathbf{c}(\mathbf{s}_0)$, but its coefficient $\boldsymbol{\Sigma}^{-1}(\mathbf{z} - \mathbf{X}\widehat{\boldsymbol{\beta}})$ is unknown. This motivates us to use a set of known nonlinear functions as the embedding of $\mathbf{s}$ in the features to characterize the spatial process $\nu(\mathbf{s})$ in the neural network. This can be done by the lights of the Karhunen–Loève (KL) theorem (Adler, 2010), which 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 nonlinear basis functions of $\mathbf{s}$.

In practice, the prediction of $\nu(\mathbf{s})$ is typically the truncated KL expansion based on the property that given any orthonormal basis functions $\phi_k(\mathbf{s})$, we can find some large integer K, so that $\nu(\mathbf{s})$ can be approximated by the finite weighted sum of basis functions, i.e., $\widehat{\nu}(\mathbf{s}) = \sum_{k=1}^{K} w_k \phi_k(\mathbf{s})$. Based on the KL theorem, the form of basis functions is not as important as the number of basis functions to approximate the spatial random effect $\nu(\mathbf{s})$. This can also be supported by the additional simulations we conduct in Section S4.1 of the Supplementary Material. Multiple types of basis functions can be used, such as the smoothing spline basis functions (Wahba, 1990), the wavelet basis functions (Vidakovic, 2009), and the radial basis functions (Friedman et al., 2001). By adding an embedding layer with sufficiently large K, the width of the neural network is greatly increased so that the network incorporates more spatial information than using the coordinates alone. A similar idea has been used in the recommendation systems by Cheng et al. (2016).

2.2 DeepKriging: a spatially dependent neural network

In this section, we use a simple DNN to illustrate our DeepKriging framework. Our model can be potentially used in other deep learning frameworks such as convolutional neural networks (CNNs) and recurrent neural

networks (RNNs).

First, we need to choose the value for K and basis functions to approximate the spatial process $\nu(\mathbf{s})$. We adopt the idea in Nychka et al. (2015), who developed a multi-resolution model for spatial prediction for large datasets. The radial basis functions at each level of resolution are constructed using a Wendland compactly supported correlation function with the nodes arranged on a rectangular grid. In particular, at a certain level of resolution, let $\{\mathbf{u}_j\}$, $j = 1, \ldots, m$, be a rectangular grid of points (or node points in the radial basis function terminology) and let θ be a scale parameter. The basis functions are given by $\phi_j^*(\mathbf{s}) = \phi(\|\mathbf{s} - \mathbf{u}_j\|/\theta)$, where

$$\phi(d) = \begin{cases} (1-d)^6(35d^2 + 18d + 3)/3, & d \in [0,1] \\ 0, & \text{otherwise.} \end{cases}$$

Hence, the embedding layer uses mutual distance locally to each knot location, implying that the spatial patterns are location invariant locally. As a result, the proposed DeepKriging is able to model the spatial non-stationarity; as we will show in Section 3.3, the induced covariance functions of an infinitely wide DeepKriging network are in general non-stationary. The scale parameter θ is set to be 2.5 times the associated knots spacing according to Nychka et al. (2015). The grid at each finer level increases by a factor of two and the basis functions are scaled to have a constant

overlap. In particular, in the h-th level, the number of knots is chosen to be $K_h = (9\times 2^{h-1}+1)^d$, where d is the spatial dimension. For a massive dataset and to obtain $K \geq N$, we need $H = 1 + \lceil \log_2(\sqrt[d]{N}/10) \rceil$ levels. Therefore, for a four-level model for instance, we need $K = 10+19+37+73 = 139$ basis functions in one dimensional space and $K = 10^2+19^2+37^2+73^2 = 7159$ basis functions in two dimensional space. This scheme gives a good approximation to standard covariance functions and also has the flexibility to fit more complicated shapes. The approach of multi-resolution approximation for massive spatial datasets has also been adopted in other research works; see Katzfuss (2017) and the references therein.

Then, for any coordinate $\mathbf{s}$, 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 recommended to be orthogonal based on the KL 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 $P + K$, and specify an L-layer DNN as

$$\begin{aligned} &\mathbf{u}_1(\mathbf{s}) = \mathbf{W}_1\mathbf{x}_\phi(\mathbf{s}) + \mathbf{b}_1, \;\; \mathbf{a}_1(\mathbf{s}) = \psi_1(\mathbf{u}_1(\mathbf{s})); \\ &\mathbf{u}_2(\mathbf{s}) = \mathbf{W}_2\mathbf{a}_1(\mathbf{s}) + \mathbf{b}_2, \;\; \mathbf{a}_2(\mathbf{s}) = \psi_2(\mathbf{u}_2(\mathbf{s})); \\ &\ldots \\ &\mathbf{u}_L(\mathbf{s}) = \mathbf{W}_L\mathbf{a}_{L-1}(\mathbf{s}) + \mathbf{b}_L, \;\; f_{\mathrm{DK}}(\mathbf{s}) = \psi_L(\mathbf{u}_L(\mathbf{s})). \end{aligned} \tag{2.4}$$

For the l-th layer with N_l neurons, $\mathbf{W}_l$ is the $N_l \times N_{l-1}$ weight matrix, $\mathbf{b}_l$ is

the bias vector of length N_l, $\mathbf{a}_l$ is the neuron vector of length N_l, and $\psi_l(\cdot)$ is the activation function. The output of this neural network is $f_{\text{DK}}(\mathbf{s})$, which is a function of the weights and biases. Let $\boldsymbol{\theta}$ be the vector of unknown weights and biases, and $\widehat{\boldsymbol{\theta}}$ be the estimate via Equation (2.2) based on the training sample. The final DeepKriging prediction at an unobserved location $\mathbf{s}_0$ is defined as $\widehat{Y}_{\text{DK}}(\mathbf{s}_0) = f_{\text{DK}}(\mathbf{s}_0; \widehat{\boldsymbol{\theta}})$.

One major advantage of our DeepKriging method 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 $N_L = 1$, $\psi_L(\cdot)$ to be an identity function, and the loss function to be the MSE. Figure 1 provides a visualization of a DeepKriging structure in two dimensional prediction for continuous data. For predicting categorical variables as in a classification problem, we choose N_L to be the number of categories, $\psi_L(\cdot)$ to be a softmax function, and the loss function to be the cross entropy loss. For the activation functions in the hidden layers, we choose the rectified linear unit (ReLU) in default, 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 the covariate effects to be spatially varying.

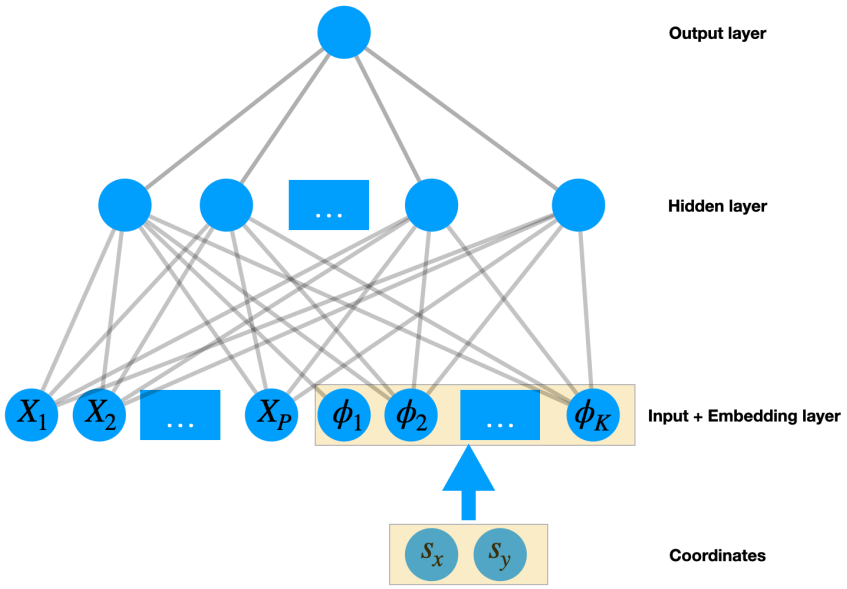

Figure 1: Visualization of the DeepKriging structure in 2D spatial prediction based on a three-layer DNN

Regularization of the DeepKriging network structure includes adding dropout layers to mitigate overfitting, adding batch-normalization layers to regularize the covariates and basis functions to the same scale, and removing all-zero columns in the basis matrix whenever they are present due to the compactly supported structure of the basis function. Details of the default setting of our DeepKriging network structure are included in Section S2 of the Supplementary Materials. The time complexity of our DeepKriging method is about $O(N_{\text{neuron}})$, where N_{neuron} is the number of neurons in the network. The computation cost depends on the width and depth of the network, and the computation is highly parallelizable and can be largely

accelerated by CPUs and GPUs.

3. Theoretical Properties of DeepKriging

DeepKriging provides a novel spatial prediction framework using deep learning. It differs from classical Kriging methods in several aspects. First, Kriging prediction is a linear combination of observations; in contrast, DeepKriging prediction is linked to the observations via the weights and biases through model training and is typically nonlinear in observations (see Section S3.1). 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.

In this section, we provide important theoretical properties of DeepKriging including 1) the underlying relationship between DeepKriging and Kriging; 2) how accurate Deepkriging can be in terms of the prediction error compared to Kriging; and 3) how the spatial dependence is measured in the DeepKriging framework. These three aspects are critical for understanding our DeepKriging method, and will be illustrated in the following subsections, respectively.

3.1 The link between DeepKriging and Kriging-based methods

DeepKriging is closely related to Kriging and its associated variants, which can be classified as multi-resolution processes (Nychka et al., 2015; Kleiber and Nychka, 2015; Katzfuss, 2017) and Gaussian predictive processes (Banerjee et al., 2008, 2010), all leading to spatial predictions that can be treated as linear functions of embedded features $\mathbf{x}_\phi(\mathbf{s}_0)$, and thus can be potentially approximated by DeepKriging.

One example is the fixed rank Kriging (FRK) proposed by Cressie and Johannesson (2008), who used 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 random effects $\nu(\mathbf{s})$ 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 also assume that the model for $Y(\mathbf{s})$ is $Y(\mathbf{s}) = \mathbf{x}(\mathbf{s})^T\boldsymbol{\beta}+\nu(\mathbf{s}) = \mathbf{x}(\mathbf{s})^T\boldsymbol{\beta}+\boldsymbol{\phi}(\mathbf{s})^T\boldsymbol{\eta}$. The covariance matrix of $Z(\mathbf{s}) = Y(\mathbf{s})+\varepsilon(\mathbf{s})$, where $\varepsilon(\mathbf{s})$ is a white noise with variance $\sigma^2(\mathbf{s})$, is given by $\boldsymbol{\Sigma} = \boldsymbol{\Phi}\boldsymbol{\Sigma}_K\boldsymbol{\Phi}^T + \boldsymbol{V}$, where $\boldsymbol{\Phi} = \{\boldsymbol{\phi}(\mathbf{s}_1), \ldots, \boldsymbol{\phi}(\mathbf{s}_N)\}^T$ is an $N \times K$ basis matrix and $\boldsymbol{V} = \text{diag}\{\sigma^2(\mathbf{s}_1), \ldots, \sigma^2(\mathbf{s}_N)\}$ is an $N \times N$ diagonal matrix. The FRK prediction as a linear function of $\mathbf{z}$ is given by

$$\widehat{Y}_{\text{FRK}}(\mathbf{s}_0) = \mathbf{x}(\mathbf{s}_0)^T\widehat{\boldsymbol{\beta}} + \boldsymbol{\phi}(\mathbf{s}_0)^T\boldsymbol{\Sigma}_K\boldsymbol{\Phi}^T\boldsymbol{\Sigma}^{-1}(\mathbf{z} - \mathbf{X}\widehat{\boldsymbol{\beta}}), \tag{3.5}$$

where $\mathbf{X} = (\mathbf{x}(\mathbf{s}_1), \cdots, \mathbf{x}(\mathbf{s}_N))^T$ is an $N \times P$ matrix, $\widehat{\boldsymbol{\beta}} = (\mathbf{X}^T\boldsymbol{\Sigma}^{-1}\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{\Sigma}^{-1}\mathbf{z}$, and $\boldsymbol{\Sigma}^{-1}$ has a computationally simple form which involves inverting the fixed rank $K \times K$ positive definite matrix $\boldsymbol{\Sigma}_K$ and the $N \times N$ diagonal matrix $\boldsymbol{V}$. Writing Equation (3.5) as $\widehat{Y}_{\text{FRK}}(\mathbf{s}_0) = \mathbf{x}(\mathbf{s}_0)^T\widehat{\boldsymbol{\beta}} + \boldsymbol{\phi}(\mathbf{s}_0)^T\widehat{\boldsymbol{\alpha}}$, where $\widehat{\boldsymbol{\alpha}} = \boldsymbol{\Sigma}_K\boldsymbol{\Phi}^T\boldsymbol{\Sigma}^{-1}(\mathbf{z} - \mathbf{X}\widehat{\boldsymbol{\beta}})$, implies that the FRK prediction $\widehat{Y}_{\text{FRK}}(\mathbf{s}_0)$ is linear in P covariates $\mathbf{x}(\mathbf{s}_0)$ and K basis functions $\boldsymbol{\phi}(\mathbf{s}_0)$. This is a special case of DeepKriging when we set all of the activation functions to be linear.

FRK usually chooses K to be much smaller than N in order to speed up the computation 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). In contrast, for DeepKriging, K needs to be sufficiently large ($K > N$) in order to have a good approximation of the spatial random effect $\nu(\mathbf{s})$, so that our method captures more spatial information in the prediction.

By setting $K = N$ in the FRK, we can see that the (universal) Kriging prediction in Equation (2.3) is also a linear function of $\mathbf{x}_\phi(\mathbf{s}_0) = (\mathbf{x}(\mathbf{s}_0)^T, \boldsymbol{\phi}(\mathbf{s}_0)^T)^T$. A detailed proof is provided in Section S1.1 in the Supplementary Materials. This result implies that the Kriging prediction with any covariance function can be linearly expressed by the embedding features $\mathbf{x}_\phi(\mathbf{s}_0)$. In this

sense, DeepKriging generalizes Kriging by allowing for nonlinear functions of $\mathbf{x}_\phi(\mathbf{s}_0)$ in the prediction.

3.2 DeepKriging in decision theory

Our DeepKriging 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 DNN model and $\widehat{Y}_N(\mathbf{s}_0)$ be the final prediction from the model based on N observed locations. The following decomposition of the total risk between the true value $Y(\mathbf{s}_0)$ and the prediction $\widehat{Y}_N(\mathbf{s}_0)$ implies three sources of errors:

$$R\{Y(\mathbf{s}_0), \widehat{Y}_N(\mathbf{s}_0)\} = \underbrace{R\{Y(\mathbf{s}_0), \widehat{Y}_{\mathcal{F}}^{\text{opt}}(\mathbf{s}_0)\}}_{\text{approximation error}} + \underbrace{R\{\widehat{Y}_{\mathcal{F}}^{\text{opt}}(\mathbf{s}_0), \widehat{Y}_N^{\text{opt}}(\mathbf{s}_0)\}}_{\text{estimation error}} + \underbrace{R\{\widehat{Y}_N^{\text{opt}}(\mathbf{s}_0), \widehat{Y}_N(\mathbf{s}_0)\}}_{\text{optimization error}}.$$

The approximation error relates to the model capacity and is defined as the risk between the true process $Y(\mathbf{s}_0)$ and the optimal predictor $\widehat{Y}_{\mathcal{F}}^{\text{opt}}(\mathbf{s}_0) = \underset{\widehat{Y}(\mathbf{s}_0)\in\mathcal{F}}{\text{argmin}}\, R(\widehat{Y}(\mathbf{s}_0), Y(\mathbf{s}_0))$ as a function in $\mathcal{F}$. The estimation error is defined as the risk between $\widehat{Y}_N^{\text{opt}}(\mathbf{s}_0)$ and $\widehat{Y}_{\mathcal{F}}^{\text{opt}}(\mathbf{s}_0)$, where $\widehat{Y}_N^{\text{opt}}(\mathbf{s}_0) = \widehat{Y}_N(\mathbf{s}_0; \hat{\boldsymbol{\theta}})$, with $\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\text{argmin}} \frac{1}{N}\sum_{n=1}^{N} L\{\widehat{Y}_N(\mathbf{s}_n; \boldsymbol{\theta}), z(\mathbf{s}_n)\}$; this type of error is affected by the complexity of $\mathcal{F}$ and relates to the generalization power of the model. The optimization error is the empirical risk between $\widehat{Y}_N^{\text{opt}}(\mathbf{s}_0)$ and $\widehat{Y}_N(\mathbf{s}_0)$.

The function class of Kriging prediction in Equation (2.3), $\mathcal{F}_{\text{UK}}$, can be viewed as the space of linear functions of $\mathbf{x}(\mathbf{s}_0)$ and $\mathbf{z}$ taking the form

$\mathbf{x}(\mathbf{s}_0)^T\boldsymbol{\beta}+\mathbf{z}^T\boldsymbol{\gamma}$, while the function class of DeepKriging, $\mathcal{F}_{\text{DK}}$, is the function space generated by the DNN described in (2.4). The universal approximation theorem (Theorem 2.3.1 of Csáji (2001)) claims that every continuous function of the features $\mathbf{x}_\phi(\mathbf{s})$, denoted as $\mathbb{C}(\mathbf{x}_\phi)$, can be arbitrarily well approximated with a feed-forward neural network with a single hidden layer that contains finite number of hidden neurons and with arbitrary activation function. This indicates that the optimal DeepKriging prediction with a single hidden layer and finite loss function has the largest model capacity in $\mathbb{C}(\mathbf{x}_\phi)$, that is, $\widehat{Y}^{\text{opt}}_{\mathcal{F}_{\text{DK}}}(\mathbf{s}_0) = \widehat{Y}^{\text{opt}}_{\mathbb{C}(\mathbf{x}_\phi)}(\mathbf{s}_0)$. This result holds for any type of data (i.e., continuous or discrete) and for any type of task (i.e., regression or classification). Therefore, the optimal DeepKriging prediction has larger capacity than the Kriging prediction in terms of minimizing the approximation error, i.e., $\mathbb{E}\{L(\widehat{Y}^{\text{opt}}_{\mathcal{F}_{\text{DK}}}(\mathbf{s}_0), Y(\mathbf{s}_0))\} \leq \mathbb{E}\{L(\widehat{Y}^{\text{opt}}_{\mathcal{F}_{\text{UK}}}(\mathbf{s}_0), Y(\mathbf{s}_0))\}$. The detailed proof is provided in Section S1.2 in the Supplementary Materials. Similarly, the optimal DeepKriging prediction also has larger model capacity than the FRK prediction. FRK can be seen as DeepKriging with a single hidden layer containing finite number of neurons and a linear activation function. By allowing for a large number of basis functions, multiple layers, more flexible activation functions and a wide network, DeepKriging yields non-linear predictions that can appropriately capture the spatial

dependence in the spatial process.

3.3 DeepKriging as a Gaussian Process

Neal (1994) showed that a single-layer fully-connected neural network with an i.i.d. prior over its parameters (i.e., weights and biases) is equivalent to a Gaussian process (GP), in the limit of infinite network width (i.e., infinite number of hidden neurons). Later, Lee et al. (2017) derived the exact equivalence between infinitely wide deep networks and GPs. Consequently, a similar correspondence to GPs also holds for our DeepKriging network.

We start from a regression-type DeepKriging model with a singe hidden layer containing N_1 neurons. The input features are $\mathbf{x}_\phi(\mathbf{s}) = (\mathbf{x}(\mathbf{s})^T, \boldsymbol{\phi}(\mathbf{s})^T)^T \in \mathbb{R}^{P+K}$, and the output is $\hat{Y}_{\text{DK}}(\mathbf{s}) = b^1 + \sum_{j=1}^{N_1} w_j^1 a_j^1(\mathbf{s})$, where $a_j^1(\mathbf{s}) = \psi_1(b_j^0 + \sum_{i=1}^{P+K} w_{ji}^0 \mathbf{x}_\phi^{(i)}(\mathbf{s}))$, with $\mathbf{x}_\phi^{(i)}(\mathbf{s})$ being the i-th component of $\mathbf{x}_\phi(\mathbf{s})$. Weights $(w_j^1,\, w_{ji}^0)$ and biases $(b^1,\, b_j^0)$ are independent and randomly drawn to have zero mean and variances σ_w^2/N_1 and σ_b^2, respectively. Consequently, the post-activations a_j^1 and $a_{j'}^1$ are independent for $j \neq j'$. Moreover, since $\hat{Y}_{\text{DK}}(\mathbf{s})$ is a sum of i.i.d terms, it follows from the Central Limit Theorem that in the limit of infinite width $N_1 \to \infty$, $\hat{Y}_{\text{DK}}(\mathbf{s})$ will be Gaussian distributed. Likewise, from the multi-dimensional Central Limit Theorem, any finite collection of $\{\hat{Y}_{\text{DK}}(\mathbf{s}_1), \hat{Y}_{\text{DK}}(\mathbf{s}_2), \cdots, \hat{Y}_{\text{DK}}(\mathbf{s}_n)\}$ will have a joint

multivariate Gaussian distribution, which is exactly the definition of a GP. Therefore, we conclude that with sufficiently large N_1, $\hat{Y}_{\text{DK}}$ is a GP with zero mean and covariance function

$$C^1(\mathbf{s}, \mathbf{s}') = E\{\hat{Y}_{\text{DK}}(\mathbf{s})\hat{Y}_{\text{DK}}(\mathbf{s}')\} = \sigma_b^2 + \sigma_w^2 E\{a_j^1(\mathbf{s})a_j^1(\mathbf{s}')\} = \sigma_b^2 + \sigma_w^2 C(\mathbf{s}, \mathbf{s}'),$$

where $C(\mathbf{s}, \mathbf{s}')$ is obtained by integrating against the distribution of w^0, b^0 as in Neal (1994).

For DeepKriging with deeper layers, the induced covariance function can be obtained in a recursive way according to Lee et al. (2017):

$$C^l(\mathbf{s}, \mathbf{s}') = \sigma_b^2 + \sigma_w^2 F_\psi(C^{l-1}(\mathbf{s}, \mathbf{s}'), C^{l-1}(\mathbf{s}, \mathbf{s}), C^{l-1}(\mathbf{s}', \mathbf{s}')), \qquad (3.6)$$

where $F_\psi(\cdot)$ is a deterministic function that depends only on the activation function ψ. An iterative series of computations can be performed to obtain the covariance C^L for the GP describing the network's final output, $\hat{Y}_{\text{DK}}(\mathbf{s})$. For the base case, $C^0(\mathbf{s}, \mathbf{s}') = \sigma_b^2 + \sigma_w^2\{\mathbf{x}_\phi(\mathbf{s})^T\mathbf{x}_\phi(\mathbf{s}')/(P + K)\}$. The aforementioned results require the assumption of infinitely many hidden neurons in each layer. However, when the prior distribution of weights and biases is Gaussian, this condition is not needed.

For certain activation functions, Equation (3.6) can be computed analytically. The simplest case occurs when the activation function is an identity function $\psi_l(x) = x$ and no covariates effect exists. Then $\hat{Y}_{\text{DK}}(\mathbf{s})$ is a

linear function of the basis functions $\boldsymbol{\phi}(\mathbf{s})$, i.e., $\hat{Y}_{\text{DK}}(\mathbf{s}) = b + \mathbf{w}^T\boldsymbol{\phi}(\mathbf{s})$, where b and $\mathbf{w}$ are combined biases and weights, respectively. In this case, the induced covariance function of $\hat{Y}_{\text{DK}}$ is given by $C^L(\mathbf{s}, \mathbf{s}') = \sigma_b^2 + \sigma_w^2\boldsymbol{\phi}(\mathbf{s})^T\boldsymbol{\phi}(\mathbf{s}')$, which is the basis approximation of a spatial covariance function.

In the case of ReLU non-linearity, Equation (3.6) has a closed form of the well-known arc-cosine kernel (Cho and Saul, 2009):

$$C^l(\mathbf{s}, \mathbf{s}') = \sigma_b^2 + \frac{\sigma_w^2}{2\pi}\sqrt{C^{l-1}(\mathbf{s}, \mathbf{s})C^{l-1}(\mathbf{s}', \mathbf{s}')}\left\{\sin\left(\theta_{\mathbf{s},\mathbf{s}'}^{l-1}\right) + (\pi - \theta_{\mathbf{s},\mathbf{s}'}^{l-1})\cos\left(\theta_{\mathbf{s},\mathbf{s}'}^{l-1}\right)\right\},$$

where $\theta_{\mathbf{s},\mathbf{s}'}^l = \cos^{-1}(C^l(\mathbf{s}, \mathbf{s}')/\sqrt{C^l(\mathbf{s}, \mathbf{s})C^l(\mathbf{s}', \mathbf{s}')})$. When no analytic form of the resulted covariance function exists, it can be computed numerically, as described in Lee et al. (2017).

Consider a regression-type DeepKriging model with a single hidden layer and no covariates effects. It can be shown that with infinitely many hidden neurons, the covariance function of the output $\hat{Y}_{\text{DK}}(\mathbf{s})$ for any two nearby locations has the form

$$C(\mathbf{s}, \mathbf{s}') = v(\mathbf{s}) + v(\mathbf{s}') - c\|\boldsymbol{\phi}(\mathbf{s}) - \boldsymbol{\phi}(\mathbf{s}')\|^2, \tag{3.7}$$

where $\boldsymbol{\phi}(\mathbf{s})$ is the basis vector at location $\mathbf{s}$, $v(\mathbf{s}) > 0$ is related to the variance when $\mathbf{s} = \mathbf{s}'$, and c is the scaling parameter. The proof is provided in Section S1.3 in the Supplementary Materials. As a special case, if only the coordinates are used in the features, then $\|\boldsymbol{\phi}(\mathbf{s}) - \boldsymbol{\phi}(\mathbf{s}')\|^2 = \|\mathbf{s} - \mathbf{s}'\|^2$,

$v(\mathbf{s}) = v(\mathbf{s}') = v$ and thus $C(\mathbf{s}, \mathbf{s}') = v - c\|\mathbf{s} - \mathbf{s}'\|^2$, which contains less information than in Equation (3.7). Therefore, the embedding layer in DeepKriging brings more flexible spatial covariance structures than simply using the coordinates.

Further, we can show how the DeepKriging induced covariance function can approximate the common stationary covariance functions in spatial statistics. Let the basis functions be $\phi_l(\mathbf{s}) = k(\mathbf{s}, \mathbf{u}_l)$ based on a certain kernel function $k(\cdot, \cdot)$ and knot $\mathbf{u}_l$, $l = 1, \cdots, K$. If the $\mathbf{u}_l$'s form a fine grid of knots covering the spatial domain, then

$$\|\boldsymbol{\phi}(\mathbf{s}) - \boldsymbol{\phi}(\mathbf{s}')\|^2 = \sum_{l=1}^{K} \{k(\mathbf{s}, \mathbf{u}_l) - k(\mathbf{s}', \mathbf{u}_l)\}^2 \approx \int \{k(\mathbf{s}, \mathbf{u}) - k(\mathbf{s}', \mathbf{u})\}^2 \mathrm{d}\mathbf{u}$$

$$= \int k(\mathbf{s}, \mathbf{u})^2 + k(\mathbf{s}', \mathbf{u})^2 - 2k(\mathbf{s}, \mathbf{u})k(\mathbf{s}', \mathbf{u}) \mathrm{d}\mathbf{u}.$$

Note that the last term is the kernel convolution approximation to a covariance function. Higdon (2002) shows that by selecting an appropriate kernel function, we can approximate any stationary covariance function based on the kernel convolution. Further, the induced covariance function of DeepKriging also possesses favorably physical interpretations. For example, DeepKriging can yield the Matérn covariance function, also commonly used in Kriging since it is related to a stochastic partial differential equation (SPDE) of Laplace type (Whittle, 1954). In addition, DeepKriging can

induce a GP that approximates a fractional Brownian motion based on the example of DNN provided in Neal (1996).

4. Simulation Studies

4.1 DeepKriging on a 1-D Gaussian process

We first consider the performance of DeepKriging when data are simulated from a 1-D stationary GP, where the Kriging prediction is optimal. We also compare DeepKriging to two naive DNNs: a DNN with only the intercept $x(s) = 1$ as the input and a DNN with $x(s) = 1$ and coordinate s as the input. We also consider Kriging prediction with the true covariance function and that with an estimated Matérn covariance function. The simulation design is illustrated in Section S3.1 of the Supplementary Materials.

Figure S1 in the Supplementary Materials shows the prediction for one of the sample datasets using each of the five prediction methods. The DNN with the intercept only predicts the mean of the process. Although including the coordinate s in the DNN improves the prediction, it fails to capture the high-frequency variability and cannot reflect the spatial correlations of the true process. Moreover, DeepKriging prediction and the optimal Kriging prediction are almost overlapped.

To further validate the performance, we calculate the root MSE (RMSE)

and mean absolute percentage error (MAPE) on the testing data over the 100 replicated samples in Table S1 in the Supplementary Material, where MAPE is defined as $\frac{1}{N_{\text{test}}} \sum_{n=1}^{N_{\text{test}}} \frac{Y_n^{\text{pred}} - Y_n^{\text{true}}}{Y_n^{\text{true}}}$, N_{test} is the number of testing samples, Y_n^{pred} is the predicted value and Y_n^{true} is the true value. As the minimum-MSE predictor, the Kriging prediction with the true covariance function has the smallest RMSE as expected. The performance of DeepKriging is comparable to the two Kriging predictions and significantly outperforms the two naive DNN models. We also provide the results on the training set in Table S1 in the Supplementary Material. Again, the Kriging prediction with the true covariance function performs the best. The DeepKriging prediction is comparable to the optimal Kriging prediction, and it outperforms the Kriging prediction with an estimated covariance function and the two naive DNN models in terms of both RMSE and MAPE.

4.2 DeepKriging on 2-D non-stationary data

In this section, we evaluate the performance of DeepKriging on 2-D non-stationary data so that the procedure is designed to resemble the real data application in Section 5. The simulation details are included in Section S3.2 of the Supplementary Material.

We use the 10-fold cross-validation method to show the performance of

DeepKriging, Kriging with an estimated stationary covariance function and the baseline DNN with only coordinates s in the features. We calculate the RMSEs and MAPEs on the testing dataset, and show the results in Figure S2(b) and Table S2. We can see that in terms of RMSE, DeepKriging significantly outperforms Kriging in terms of RMSEs and MAPEs, since Kriging assumes a stationary covariance function while DeepKriging captures the non-stationarity in the data. In addition, the baseline DNN is better than Kriging in this example because the data are non-Gaussian and Kriging is no longer optimal. Moreover, the baseline DNN performs worse than DeepKriging as expected. The MAPE from DeepKriging is lower than the baseline DNN but higher than Kriging; this can happen since we are using MSE as the loss function in DeepKriging so it not necessarily possesses the lowest MAPE. We also calculate the RMSEs and MAPEs on the training dataset (see Table S2). Kriging outperforms the other two models in terms of both metrics. This is because the errors for the training dataset can be viewed as the variance estimates of the assumed model, similar as in a regression model. Kriging tends to underestimate such a variance, leading to a worse prediction on the testing dataset.

Additional simulations (see Section S3 of the Supplementary Material) are conducted to show that DeepKriging is non-linear in observation

whereas Kriging is linear. Furthermore, the comparison of computation time based on the same simulation study shows that Kriging is faster for small sample sizes ($N < 1,500$), but DeepKriging is much more scalable when the sample size increases. This is because when the sample size is small, the computation time is still under control for Kriging, but for DeepKriging, the number of parameters is large due to the large width and depth of the network, making the computation time longer than Kriging. When the sample size increases, the computational burden of both methods also increases, but for DeepKriging, we can use parallel computing for the data with CPUs or GPUs to largely accelerate the computation. Therefore, our DeepKriging method is much more scalable to large data sizes. For example, when $N = 12,800$, it takes more than 1.5 hours ($5,663$ seconds) to implement a Kriging model, while DeepKriging only takes 3.5 minutes (214 seconds) without GPU acceleration and 1.5 minutes (94 seconds) with a Tesla P100 GPU.

5. Application

5.1 Challenges of predicting $PM_{2.5}$ concentration

$PM_{2.5}$, fine particulate matter of less than $2.5\,\mu m$, is a harmful air pollutant. Its adverse effects are associated with many diseases such as respi-

ratory disease (Peng et al., 2009) and myocardial infarction (Peters et al., 2001); see the review by World Health Organization (2013). Therefore, it is essential to obtain a high-resolution map of $PM_{2.5}$ exposure in order to assess its impact. The measurements from monitoring networks are the best characterization of $PM_{2.5}$ concentration at a given time and location. However, data from monitoring locations are often sparsely distributed so that they are out of spatial and temporal alignment with health outcomes. Meanwhile, it is known that $PM_{2.5}$ concentration 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 with good spatial coverage and resolutions. Hence, the interpolation of $PM_{2.5}$ concentration by making use of data from monitoring networks and other meteorological data has been a promising field of research (Di et al., 2016), where spatial prediction plays a central role.

The modeling and prediction of $PM_{2.5}$ concentration are challenging. First, $PM_{2.5}$ concentration data are obviously non-Gaussian, and thus classical Kriging methods are inappropriate here. Second, $PM_{2.5}$ data from monitoring stations are irregular and sparse, but many interpolation methods require lattice data. Third, it is more important but challenging to understand the risk of high pollution and predict pollution levels, such as

being low, medium and high; statistically, these two questions are related to estimating the probability over a threshold and a classification problem, respectively. Quantile regression and convolutional neural networks have been employed to overcome some of the above issues (Reich et al., 2011; Porter et al., 2015; Di et al., 2016). However, a unified method to handle all of the aforementioned tasks has not yet been sufficiently developed.

5.2 Data and preprocessing

To tackle the above-mentioned problems, we apply the proposed DeepK-riging method to the spatial prediction of $PM_{2.5}$ concentrations based on meteorological variables. Meteorological data are obtained from the NCEP North American Regional Reanalysis (NARR) product. Reanalysis is a gridded dataset that represents the state of the atmosphere, incorporating observations and outputs of numerical weather prediction models from past to present-day. Reanalysis data are often used to represent the "true state" of the atmosphere according to observations, and thus we use the reanalysis data as the "observed data" for the covariates. 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. The covariates from the NARR product are gridded data on June 05, 2019 with a spatial resolution of about 32×32 km covering the continental U.S., containing $7,706$ gridded cells in total. Since the units of the meteorological variables are different, we use min-max normalization to re-scale the data before implementing the models. Daily averaged data of $PM_{2.5}$ concentrations are observed from 841 monitoring stations. Since the coordinates from NARR and those from stations are not identical and some of stations are too close to each other, we keep the spatial resolution of NARR and average the $PM_{2.5}$ measurements of nearby monitoring stations in the same grid cell. After the matching, 604 grid cells remain for the model training, with the $PM_{2.5}$ concentration value at each location shown in Figure 2(a). Our goal is to predict the $PM_{2.5}$ concentrations at any s_0 of the other $7,706 - 604 = 7,102$ locations where the $PM_{2.5}$ concentrations are not observed but the covariates are provided by the reanalysis data.

5.3 Model fitting and results

Our aim is to predict the $PM_{2.5}$ concentration values at unobserved grid cells where the six meteorological variables are provided. 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 with

the six covariates and coordinates. We calculate the MSEs and MAEs as the validation criterion, shown in the first two rows of Table 1, which imply that DeepKriging outperforms the baseline DNN and Kriging.

To assess the risk of high $PM_{2.5}$ pollution, we can use DeepKriging for spatial data classification. Specifically, we threshold the $PM_{2.5}$ concentrations by 12.0 $\mu g/m^3$, which is the threshold between "good" and "moderate" levels for the daily mean of EPA national ambient air quality standards (NAAQS) (EPA, 2012). Based on the classified data, we can implement a binary classification with DeepKriging by assuming the actual values of $PM_{2.5}$ concentration to be unknown. A direct comparison to Kriging is not feasible since Kriging is not suitable for binary classification. Instead, we predict the continuous $PM_{2.5}$ concentrations using Kriging and then classify the predictions by thresholding them at 12.0 $\mu g/m^3$. We then use the 10-fold cross-validation to show the classification accuracy, presented in the last row of Table 1. We can see that DeepKriging significantly outperforms Kriging and baseline DNN in terms of the classification accuracy.

Based on the model fitting, we can further predict the value of $PM_{2.5}$ concentration, the level of pollution and the risk of high pollution level over the threshold 12 $\mu g/m^3$ at unobserved locations based on the NARR data. Figure 2(a) shows the raw $PM_{2.5}$ station data from the AQS database. Fig-

Table 1: Model performance based on the 10-fold cross-validation. MSEs and MAEs of the predictions, as well as classification accuracy (ACC) for predicting $PM_{2.5}$ concentrations above 12.0 $\mu g/m^3$ are used as the validation criteria. Mean and standard deviation (SD) of the 10 sets of validation errors or accuracy are provided in the table.

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%

ure 2(b) shows a smooth map of the predicted $PM_{2.5}$ concentration from DeepKriging. We also provide the distribution prediction (details and algorithms are included in Section S5 in the Supplementary Material) in order to obtain the predicted risk defined as $\mathbb{P}\{PM_{2.5} > 12\ \mu g/m^3\}$, shown in Figure 2(c). This map implies that high $PM_{2.5}$ pollution risks exist over a vast area of Eastern US. We further compare the results to the Kriging prediction in Figure 2(d), which implies that DeepKriging provides more local features/patterns than Kriging.

6. Discussion

In this work, we have proposed a new spatial prediction model using deep neural networks which incorporates the spatial dependence by a set of ba-

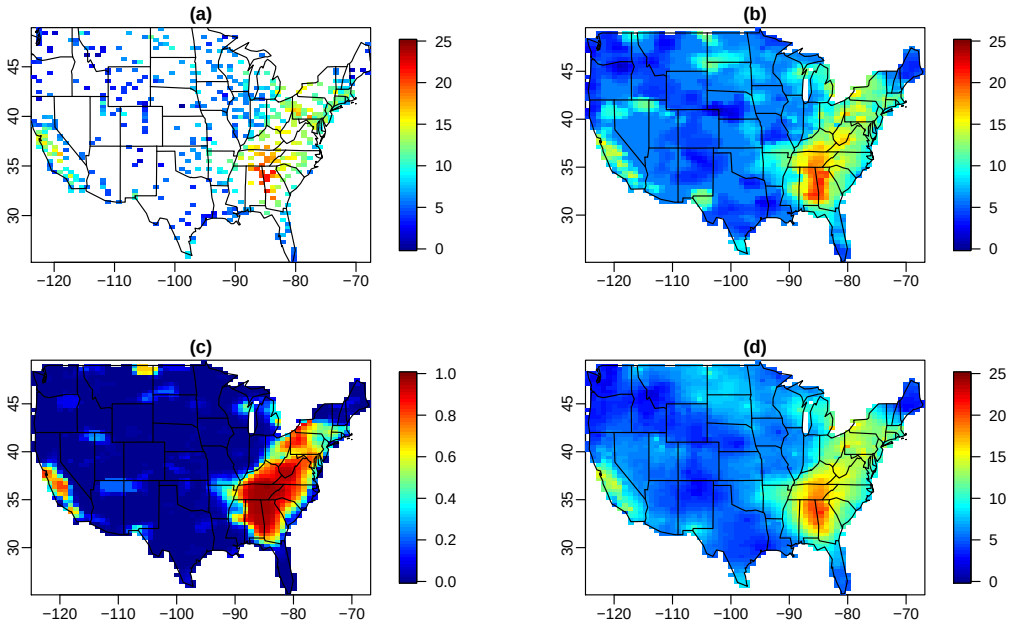

Figure 2: (a) $PM_{2.5}$ concentration ($\mu g/m^3$) collected from monitoring stations. (b) Predicted $PM_{2.5}$ concentration using DeepKriging. (c) Predicted risk of high pollution $\mathbb{P}\{PM_{2.5} > 12\ \mu g/m^3\}$ based on distribution prediction using DeepKriging. (d) Predicted $PM_{2.5}$ concentration using Kriging.

sis 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. Uncertainty quantification can be provided based on our DeepKriging framework using the distribution prediction method detailed in Section S5 in the Supplementary Materials.

Classical Kriging methods consider their predictions as linear combinations 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 in spatial prediction with a large number of basis functions. We have shown that the proposed method is superior to Kriging in many aspects both theoretically and numerically in our simulation and real application. For instance, DeepKriging is more scalable for large datasets and suits for more data types than Kriging. DeepKriging also has a GP representation with flexible spatial covariance structures, which enables Bayesian inference on regression tasks by evaluating the corresponding GP. More importantly, the proposed DeepKriging framework connects the regression-based prediction and spatial prediction so that many other machine learning algorithms can be applied.

In general applications, it is possible that the covariates at the new location $\mathbf{s}_0$ are not observed. One promising approach for coping with this problem is to find the true values of the missing covariates for a subset of the observations and then train a machine learning algorithm to predict the

values of those covariates for the rest (see, e.g., Imai and Khanna (2016)). However, Fong and Tyler (2021) showed that plugging in these predictions without regard for prediction error renders regression analyses biased, inconsistent, and overconfident. They described a procedure to avoid these inconsistencies. This approach combines a new sample splitting scheme and a general method of moments (GMM) estimator to make an efficient and consistent estimator. Overall, it is non-trivial to address the problem of missing covariates: intuitive strategies such as plugging in machine learning predictions lead to bias and inconsistency, while the implementation of a more complicated method such as that in Fong and Tyler (2021) requires extra assumptions (e.g., the exclusion restriction condition) and increases computational burden. If the goal is to predict both the response and the covariates instead, a multivariate version of DeepKriging could be developed. These are left as our future work.

Supplementary Materials

The Supplementary Material contains details referenced in the main manuscript, including the proofs of the lemmas and theorems (Section S1), the settings for the DeepKriging network structure (Section S2), details of the simulation studies (Section S3), additional simulation studies (Section S4), dis-

tribution prediction and uncertainty quantification (Section S5), and the source codes and data for reproducible research (Section S6).

Acknowledgments

This research is supported by the National Key Research and Development Program (2021YFA1000101), Zhejiang Provincial Natural Science Foundation of China (LZJWY22E090009), Natural Science Foundation of Shanghai (22ZR1420500), the Open Research Fund of Key Laboratory of Advanced Theory and Application in Statistics and Data Science-MOE, ECNU and King Abdullah University of Science and Technology (KAUST), Office of Sponsored Research (OSR) under Award No: OSR-2019-CRG7-3800.

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 modeling. *Ecological Modeling*, 157(2-3):101–118.

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.

Cheng, H.-T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., et al. (2016). Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*, pages 7–10.

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*, 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.

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

Fong, C. and Tyler, M. (2021). Machine learning predictions as regression covariates. *Political Analysis*, 29(4):467–484.

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.

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

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.

Imai, K. and Khanna, K. (2016). Improving ecological inference by predicting individual ethnicity from voter registration records. *Political Analysis*, 24(2):263–272.

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

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

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., Sohl-Dickstein, J., Pennington, J., Novak, R., Schoenholz, S., and Bahri, Y. (2018). Deep neural networks as Gaussian processes. *International Conference on Learning Representations*.

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

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. (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. (1994). Priors for infinite networks. *Technical Report*.

Neal, R. M. (1996). *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.

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.

Reich, B. J., Fuentes, M., and Dunson, D. B. (2011). 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.

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.

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

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

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

Wanfang Chen: Academy of Statistics and Interdisciplinary Sciences, East China Normal University, Shanghai, China

E-mail: wfchen@fem.ecnu.edu.cn

Yuxiao Li: Statistics Program, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia

E-mail: yuxiao.li@kaust.edu.sa

Brian J Reich: Department of Statistics, North Carolina State University, Raleigh, North Carolina, U.S.A.

E-mail: brian_reich@ncsu.edu

Ying Sun (corresponding author): Statistics Program, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia

E-mail: ying.sun@kaust.edu.sa; Phone: +966 (0) 56 898-5402; Tax: +966 (0) 12 808-0644

Supplementary Material

This supplementary material contains the proofs (Section S1), the settings for the DeepKriging network structure (Section S2), details of the simulation studies (Section S3), additional simulation studies (Section S4), distribution prediction and uncertainty quantification (Section S5), and the source codes and data for reproducible research (Section S6).

S1. Proofs

S1.1 Kriging prediction is linear in $\mathbf{x}_\phi(\mathbf{s}) = (\mathbf{x}(\mathbf{s}), \boldsymbol{\phi}(\mathbf{s}))^T$

Proof: Let the covariance matrix associated with the random process $\nu(\mathbf{s})$ be $\boldsymbol{\Sigma}^\nu$, where $\boldsymbol{\Sigma}^\nu_{i,j} = C(\mathbf{s}_i, \mathbf{s}_j)$. We can build a spatial random effect process $\mu(\mathbf{s}) = \boldsymbol{\phi}(\mathbf{s})^T\boldsymbol{\eta}$ with $\text{Cov}(\boldsymbol{\eta}) = \boldsymbol{\Phi}_R^{-1}\boldsymbol{\Sigma}^\nu(\boldsymbol{\Phi}_R^{-1})^T$, where $\boldsymbol{\Phi} = \{\boldsymbol{\phi}(\mathbf{s}_1), \ldots, \boldsymbol{\phi}(\mathbf{s}_N)\}$ is an $N \times K$ basis matrix with rank N, and $\boldsymbol{\Phi}_R^{-1}$ is the right inverse of $\boldsymbol{\Phi}$. From the result of Banerjee (1973), the right inverse exists. Hence, $\text{Cov}(\boldsymbol{\eta})$ is also a valid covariance matrix and $\text{Cov}(\boldsymbol{\mu}) = \text{Cov}(\boldsymbol{\nu}) = \boldsymbol{\Sigma}^\nu$, 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 $\nu(\mathbf{s})$, i.e., $\widehat{Y}_\mu^{\text{UK}}(\mathbf{s}_0) = \widehat{Y}_\nu^{\text{UK}}(\mathbf{s}_0)$. 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 (5) in the manuscript, such that $\widehat{Y}_\mu^{\text{UK}}(\mathbf{s}_0) = \widehat{Y}_\mu^{\text{FRK}}(\mathbf{s}_0) = \mathbf{x}(\mathbf{s}_0)^T\widehat{\boldsymbol{\beta}} + \boldsymbol{\phi}(\mathbf{s}_0)^T\widehat{\boldsymbol{\alpha}}$. $\square$

S1.2 Model capacity of DeepKriging compared to Kriging

Proof: 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 $\forall f \in \mathbb{C}(\mathbf{x}_\phi)$, $\forall \varepsilon > 0$: $\exists n \in \mathbb{N}$ 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 multi-layer feed-forward networks with a single hidden layer that contains n hidden neurons. Then we have

$$\lim_{n\to\infty} |f(\mathbf{s}_0) - A_n^{\text{opt}} f(\mathbf{s}_0)|^2 = 0, \tag{S1.1}$$

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 $\mathbb{C}(\mathbf{x}_\phi)$ is $\widehat{Y}^{\text{opt}}_{\mathbb{C}(\mathbf{x}_\phi)}(\mathbf{s}_0)$ that minimizes the mean squared loss, i.e., $\widehat{Y}^{\text{opt}}_{\mathbb{C}(\mathbf{x}_\phi)}(\mathbf{s}_0) = \underset{\widehat{Y}(\mathbf{s}_0)\in\mathbb{C}(\mathbf{x}_\phi)}{\text{argmin}}\ \mathbb{E}\{|\widehat{Y}(\mathbf{s}_0) - Y(\mathbf{s}_0)|^2\}$. Then, the mean squared loss of $\widehat{Y}^{\text{opt}}_{\mathcal{F}_{\text{DK}}}(\mathbf{s}_0)$ satisfies

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

Let $f(\mathbf{s}_0) = \widehat{Y}^{\text{opt}}_{\mathbb{C}(\mathbf{x}_\phi)}(\mathbf{s}_0) \in \mathbb{C}(\mathbf{x}_\phi)$ in (S1.1), then $A_n^{\text{opt}} f(\mathbf{s}_0) = \widehat{Y}^{\text{opt}}_{\mathcal{F}_{\text{DK}}}(\mathbf{s}_0)$ based on the definition, hence $\lim_{n\to\infty} |\widehat{Y}^{\text{opt}}_{\mathcal{F}_{\text{DK}}}(\mathbf{s}_0) - \widehat{Y}^{\text{opt}}_{\mathbb{C}(\mathbf{x}_\phi)}(\mathbf{s}_0)|^2 = 0$. Also note that

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

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

Since $\mathcal{F}_{\text{UK}} \subset \mathbb{C}(\mathbf{x}_\phi)$, we have

$$\mathbb{E}\{L(\widehat{Y}^{\text{opt}}_{\mathcal{F}_{\text{DK}}}(\mathbf{s}_0), Y(\mathbf{s}_0))\} = \mathbb{E}\{L(\widehat{Y}^{\text{opt}}_{\mathbb{C}(\mathbf{x}_\phi)}(\mathbf{s}_0), Y(\mathbf{s}_0))\} \le \mathbb{E}\{L(\widehat{Y}^{\text{opt}}_{\mathcal{F}_{\text{UK}}}(\mathbf{s}_0), Y(\mathbf{s}_0))\}. \quad \square$$

S1.3 Proof of Equation (3.7)

Proof: Recall that the induced covariance function of $\hat{Y}_{\mathrm{DK}}(\mathbf{s})$ is

$$C(\mathbf{s},\mathbf{s}') = \mathbb{E}\{Y(\mathbf{s})Y(\mathbf{s}')\} = \sigma_b^2 + \sigma_w^2\mathbb{E}\{a_j^1(\mathbf{s})a_j^1(\mathbf{s}')\}.$$

According to Neal (1994),

$$\mathbb{E}\{a_j^1(\mathbf{s})a_j^1(\mathbf{s}')\} = \frac{1}{2}[\mathbb{V}\mathrm{ar}\{a_j^1(\mathbf{s})\} + \mathbb{V}\mathrm{ar}\{a_j^1(\mathbf{s}')\}] - \frac{1}{2}\mathbb{E}\{[a_j^1(\mathbf{s}) - a_j^1(\mathbf{s}')]^2\}.$$

When $\mathbf{s}$ is close to $\mathbf{s}'$, we have $\psi_1(x) - \psi_1(y) = \alpha(x-y)$ for a smooth activation function, where α is a scaling coefficient. When the covariate vector $\mathbf{x}(\mathbf{s})$ is not available, we have $a_j^1(\mathbf{s}) = \psi_1(b_j^0 + \sum_{k=1}^K w_{jk}^0\phi_k(\mathbf{s}))$. Thus we have

$$\mathbb{E}\{[a_j^1(\mathbf{s}) - a_j^1(\mathbf{s}')]^2\} = (\alpha\sigma_w)^2\sum_{k=1}^K\{\phi_k(\mathbf{s}) - \phi_k(\mathbf{s}')\}^2.$$

Finally, the covariance function for nearby locations is

$$C(\mathbf{s},\mathbf{s}') = \sigma_b^2 + \frac{\sigma_w^2}{2}[\mathbb{V}\mathrm{ar}\{a_j^1(\mathbf{s})\} + \mathbb{V}\mathrm{ar}\{a_j^1(\mathbf{s}')\}] - (\alpha\sigma_w^2)^2\sum_{k=1}^K\{\phi_k(\mathbf{s}) - \phi_k(\mathbf{s}')\}^2$$

$$\equiv v(\mathbf{s}) + v(\mathbf{s}') - c\|\boldsymbol{\phi}(\mathbf{s}) - \boldsymbol{\phi}(\mathbf{s}')\|^2. \quad \square$$

S2. Settings for the DeepKriging network structure

The model settings for the DeepKriging network structure include the following considerations. First, when the sample size is small, we add a dropout layer to avoid overfitting. For a large dataset with Gaussian priors, dropout and other types of regularization are not very helpful. Second, batch-normalization is another useful strategy since the covariates typically have different units, and the scales of basis function may vary too much for

irregularly spaced spatial data. Third, we normalize the covariates before we run the automated batch-normalization since the covariates and basis functions required different types of normalization. Last, when the data are irregularly spaced, the value of the compactly supported basis functions for some knots that are far apart from any other locations will be zero; in this case, we remove these knots and the relevant columns in the basis matrix.

To summarize, the default setting of DeepKriging network is as follows:

(1) Normalize (min-max normalization) the observed covariates $\mathbf{x}(\mathbf{s})$;

(2) Build the embedding layer using a 3 to 5 level multi-resolution radial basis functions $\boldsymbol{\phi}(\mathbf{s})$ with the corresponding basis matrix $\boldsymbol{\Phi}$; the form of the basis function and the choice of the number of basis functions K are illustrated at the beginning of Section 2.2;

(3) Remove the all-zero columns of the basis matrix $\boldsymbol{\Phi}$;

(4) Add the 1st dense layer with 100 hidden neurons and ReLu activation;

(5) Add the 1st dropout layer with 0.5 dropout rate;

(6) Add the 1st batch-normalization layer;

(7) Add the 2nd dense layer with 100 hidden neurons and ReLu activation;

(8) Add the 2nd dropout layer with 0.5 dropout rate;

(9) Add the 3rd dense layer with 100 hidden neurons and ReLu activation;

(10) Add the 2nd batch-normalization layer;

(11) Add the output layer.

The default number of epochs for DeepKriging is 200. For regression tasks, the loss function is set to be the mean squared error (MSE), while for classification tasks, the loss function is set to be the cross-entropy loss. For optimization, we use the Adam optimizer (Kingma and Ba, 2014). One can tune the hyper-parameters in the network (e.g., the value of K, the number of layers, the width of each layer, the dropout rate, the batch size and the number of epochs) in order for the optimization to converge and to achieve a better performance compared to other basic methods such as Kriging.

S3. Details of Simulation Studies

S3.1 DeepKriging on a 1-D Gaussian process

We first consider the performance of DeepKriging when data are simulated from a 1-D stationary GP, where the Kriging prediction is optimal. We also compare DeepKriging (with $x(s)$ and an embedding layer with basis functions of coordinate s as the input) to a DNN that does not account for the spatial dependence (with only $x(s)$ as the input), and a DNN that incorporates the spatial dependence by including directly the coordinates in the features (with $x(s)$ and coordinate s as the input). Specifically, the 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 $C(s, s') = \sigma^2 \exp\{-|s - s'|/\rho\}$, where the variance $\sigma^2 = 1$ and the range parameter $\rho = 0.1$, and $\varepsilon(s)$ is a Gaussian white noise with the nugget vari-

ance $\tau^2 = 0.01$. We generate 100 replicates for $\{z(s_1), \cdots, z(s_N)\}$ from the GP with $N = 1,000$ equally spaced locations over $[0, 1]$, with 800 locations randomly selected as training data and the remaining treated as testing data. In this example, there are no observed covariates except for the intercept, i.e., $x(s) = 1$ for any $s \in \mathbb{R}$. We also compare these models to Kriging prediction with the true exponential covariance function and that with an estimated Matérn covariance function where the smoothness parameter is set to 1.5. The above predictions related to Gaussian processes and deep learning are implemented using GPy (GPy, 2012) and Keras (Ketkar et al., 2017), respectively. Since we have Gaussian priors, dropout and other types of regularization for DeepKriging are not needed. The number of hidden layers is set to be 7 in DeepKriging, which yields the best performance (in terms of RMSE) for both the training and testing datasets among networks with 1 to 10 hidden layers (results not shown). Each hidden layer contains 100 neurons and uses the ReLu activation. The loss function is MSE, the number of epochs is set to be 100 and the batch size is set to be 32. A four-level multi-resolution model is used to generate the basis functions for DeepKriging, thus $K = 10 + 19 + 37 + 73 = 139$.

Figure S1 shows the prediction for one of the sample datasets using each of the five prediction methods. Table S1 shows the root mean squared error (RMSE) and mean absolute percentage error (MAPE) on the both the training and testing sets over the 100 replicated samples.

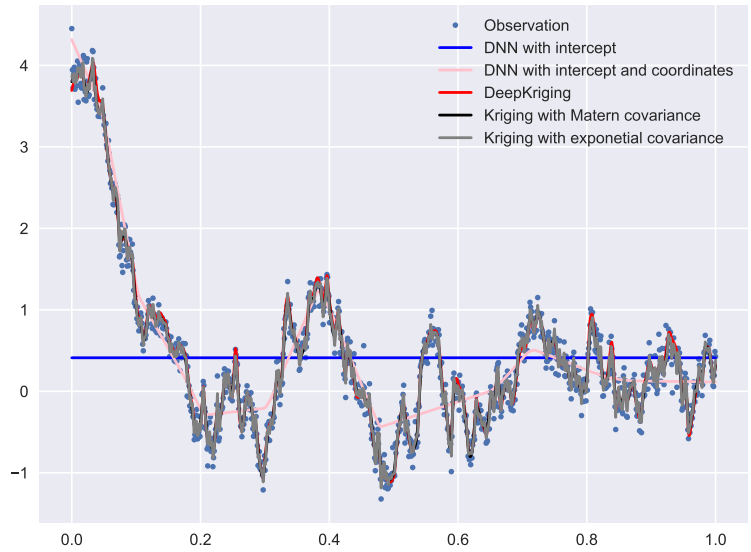

Figure S1: Prediction results for one simulated dataset generated from a GP. The blue dots are the simulated data (observations). The solid lines are the prediction curves from DNN with intercept only (blue line), DNN with intercept and coordinates (orange line), DeepKriging (red line), Kriging with the true exponential covariance function (grey line), and Kriging with an estimated Matérn covariance function (black line), respectively.

S3.2 DeepKriging on 2-D non-stationary data

In this section, we evaluate the performance of DeepKriging on 2-D non-stationary data so that the procedure is designed to resemble the real data application in Section 5 of the main manuscript. The goal is to predict values from the true process: $Y(\mathbf{s}) = \sin\{30(\bar{s}-0.9)^4\}\cos\{2(\bar{s}-0.9)\} + (\bar{s}-0.9)/2$, where $\mathbf{s} = (s_x, s_y)^T \in \mathbb{R}^2$ and $\bar{s} = (s_x + s_y)/2$. A similar example is evaluated in many computer experiments (Ba et al., 2012), where both Kriging and neural networks are popularly applied. In our simulation,

Table S1: Mean and standard deviations (SD) of the root mean squared errors (RMSEs) and mean absolute percentage errors (MAPEs) on both the training and testing sets across the 100 datasets from the five predictions of a Gaussian process. Kriging I and II are the Kriging prediction with the true exponential covariance function and that with an estimated Matérn covariance function, respectively. DeepKriging prediction is based on the MSE loss. DNN I and DNN II are the DNN prediction with intercept only and that with intercept and coordinates, respectively.

Models	Kriging I		Kriging II		DeepKriging		DNN I		DNN II	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Training set										
RMSE	**.068**	.002	.125	.006	.083	.009	.887	.183	.280	.029
MAPE	**.431**	1.639	.890	3.571	.851	3.834	5.158	9.538	1.779	4.475
Testing set										
RMSE	**.158**	.008	.164	.008	.258	.015	.885	.185	.292	.031
MAPE	**.536**	.504	.570	.548	.638	.540	3.123	2.755	.942	.843

$N = 900$ observations are sampled on a $30\times$ 30 square grid of locations spanning $[0, 1]^2$; see Figure S2(a). This process presents obvious spatial non-stationarity; for example, the smoothness over the region $[0, 0.4]^2$ is significantly smaller than that over $[0.4, 1]^2$.

The network structures of DeepKriging as well as the baseline DNN are the default setting as stated in Section S2, except that the batch size is set to be 64, and the number of hidden layers is set to be 4 in DeepKriging, which yields the best performance (in terms of RMSE) for both the training and testing datasets among networks with 1 to 6 hidden layers (results not shown). A three-level multi-resolution model is used to generate the basis functions for DeepKriging, thus $K = 10^2 + 19^2 + 37^2 = 1,830$. We use the 10-fold cross-validation method to show the performance of DeepKriging,

Kriging with an estimated stationary covariance function and the baseline DNN with only coordinates s in the features. The boxplot of RMSEs are shown in Figure S2(b). The RMSEs and MAPEs on both the training and testing sets are shown in Table S2.

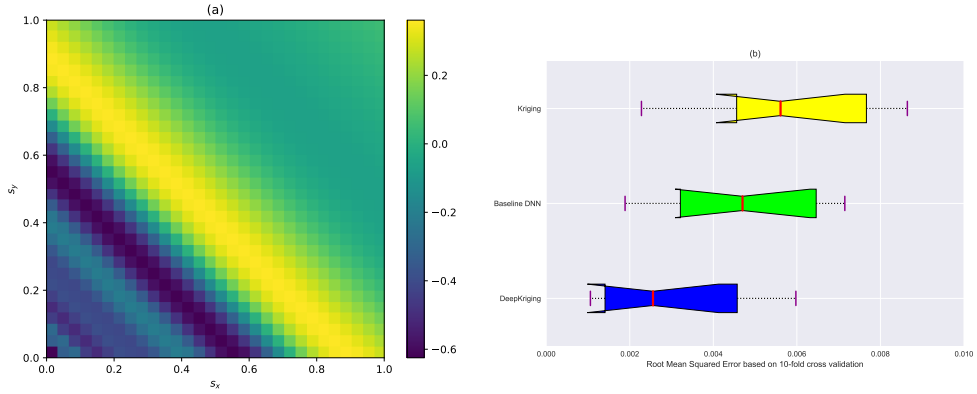

Figure S2: (a) Visualization of the simulated data generated from $Y(\mathbf{s}) = \sin\{30(\bar{s}-0.9)^4\}\cos\{2(\bar{s}-0.9)\} + (\bar{s}-0.9)/2$, where $\mathbf{s} = (s_x, s_y)^T \in [0,1]^2$ and $\bar{s} = (s_x + s_y)/2$. (b) Boxplots of the 10 RMSEs based on the 10-fold cross-validations from DeepKriging (blue), baseline DNN (green), and Kriging (yellow).

S4. Additional Simulations

S4.1 Choice of basis functions

As we have stated in the main manuscript, based on the KL theorem, the form of basis functions is not as important as the number of basis functions to approximate the spatial random effect $\nu(\mathbf{s})$. This can be supported by the additional simulations we conduct below.

Table S2: Model performance for both the training and testing sets based on 10-fold cross-validation. Data are generated from $Y(\mathbf{s}) = \sin\{30(\bar{s} - 0.9)^4\}\cos\{2(\bar{s} - 0.9)\} + (\bar{s} - 0.9)/2$, where $\mathbf{s} = (s_x, s_y)^T \in [0,1]^2$ and $\bar{s} = (s_x + s_y)/2$. RMSEs and MAPEs are computed, and the mean and standard deviations (SD) for the 10 sets of validation errors are given. DeepKriging prediction is based on the MSE loss. The baseline DNN includes only coordinates $\mathbf{s}$ in the features. The Kriging prediction uses an estimated exponential covariance function.

Models	DeepKriging		Baseline DNN		Kriging	
	Mean	SD	Mean	SD	Mean	SD
Training set						
RMSE ($\times 10^{-3}$)	**.585**	.869	4.255	2.384	5.969e-4	7.287e-5
MAPE	**1.269**	1.313	5.296	3.170	5.749e-7	5.200e-8
Testing set						
RMSE ($\times 10^{-3}$)	**3.466**	3.417	6.934	7.672	8.552	9.562
MAPE	**5.330**	3.885	6.152	3.221	0.007	0.005

In Nychka et al. (2015), they stated that "the choice of the Wendland family of RBFs is not crucial and other compactly supported, positive definite functions will work." They used compactly supported kernels so that the key matrices are sparse to improve the computational efficiency. Our DeepKriging method does not need matrix operations, hence in principle we could use any positive definite kernels. For the 1-D simulation study of Section 4.1, here we replace our original choice of the Wendland kernel with a Gaussian kernel. Specifically, at a certain level of resolution, let $\{\mathbf{u}_j\}$, $j = 1, \ldots, m$, be a rectangular grid of points (i.e., the node points), and let θ be a scale parameter. The basis functions are then given by $\phi_j^*(\mathbf{s}) = \phi(\|\mathbf{s} - \mathbf{u}_j\|/\theta)$, where $\phi(d) = \exp\{-d^2\}$. We use the same ex-

perimental designs for the 1-D simulation as before. Figure S3 shows the prediction for one of the sample datasets using each of the six prediction methods. DeepKriging prediction with both basis function choices almost overlap with the optimal Kriging prediction. We then compare the RMSEs and MAPEs on both the training and testing datasets, as shown in Table S3 below. The results for predictions other than DeepKriging II (with the new basis functions) are only slightly different from our original results in Table S1, and the results are very close to each other between the two DeepKriging predictions. In addition, changing the basis functions do not change the conclusions overall; that is, for the testing set, the Kriging prediction with the true covariance function has the smallest RMSE as expected, and the performance of DeepKriging is comparable to the two Kriging predictions and outperforms the two naive DNN models. For the training set, both DeepKriging predictions are comparable to the optimal Kriging prediction, and they outperform the Kriging prediction with an estimated covariance function and the two naive DNN models in terms of both RMSE and MAPE.

Similarly for the 2-D simulation study in Section 4.2, replacing the radial basis functions using Wendland kernel with those using Gaussian kernel does not change the conclusion, if we compare Figure S4(b) with Figure S2(b), and compare Table S4 with Table S2. Specifically, for the testing set, in terms of RMSE, both DeepKriging predictions significantly outperforms Kriging in terms of RMSEs and MAPEs. In addition, the

baseline DNN is better than Kriging because the data are non-Gaussian and Kriging is no longer optimal. Moreover, the baseline DNN performs worse than DeepKriging as expected. The MAPE from DeepKriging is lower than the baseline DNN but higher than Kriging; this can happen since we are using MSE as the loss function in DeepKriging so it not necessarily possesses the lowest MAPE. For the training set, Kriging performs best in terms of both metrics since Kriging tends to underestimate such a variance, leading to a worse prediction on the testing dataset.

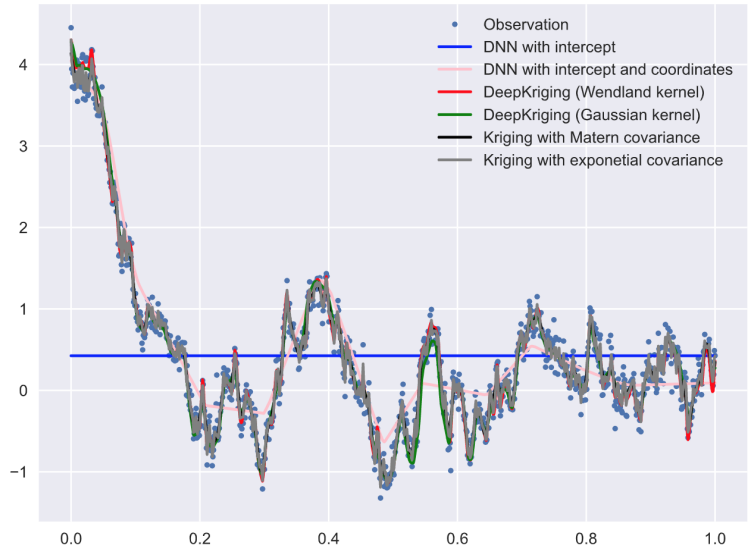

Figure S3: Prediction results for one simulated dataset generated from a GP. The blue dots are the simulated data (observations). The solid lines are the prediction curves from DNN with intercept only (blue line), DNN with intercept and coordinates (orange line), DeepKriging with Wenland kernel basis functions (red line), DeepKriging with Gaussian kernel basis functions (green line), Kriging with the true exponential covariance function (grey line), and Kriging with an estimated Matérn covariance function (black line), respectively.

Table S3: Mean and standard deviations (SD) of the root mean squared errors (RMSEs) and mean absolute percentage errors (MAPEs) on both the training and testing sets across the 100 datasets from the six predictions of a Gaussian process. Kriging I and II are the Kriging prediction with the true exponential covariance function and that with an estimated Matérn covariance function, respectively. DeepKriging I and II are the DeepKriging prediction with Wenland kernel and Gaussian kernel, respectively, based on the MSE loss. DNN I and DNN II are the DNN prediction with intercept only and that with intercept and coordinates, respectively.

Models	Kriging I		Kriging II		DeepKriging I		DeepKriging II		DNN I		DNN II	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Training set												
RMSE	**.063**	.002	.125	.006	.087	.011	.120	.008	.887	.183	.243	.025
MAPE	**.217**	.128	.456	.318	.274	.189	.450	.453	4.030	4.725	.915	.921
Testing set												
RMSE	**.160**	.008	.165	.008	.197	.018	.187	.015	.884	.186	.254	.029
MAPE	**.573**	.541	.603	.583	.676	.639	.676	.635	3.495	2.747	.880	.781

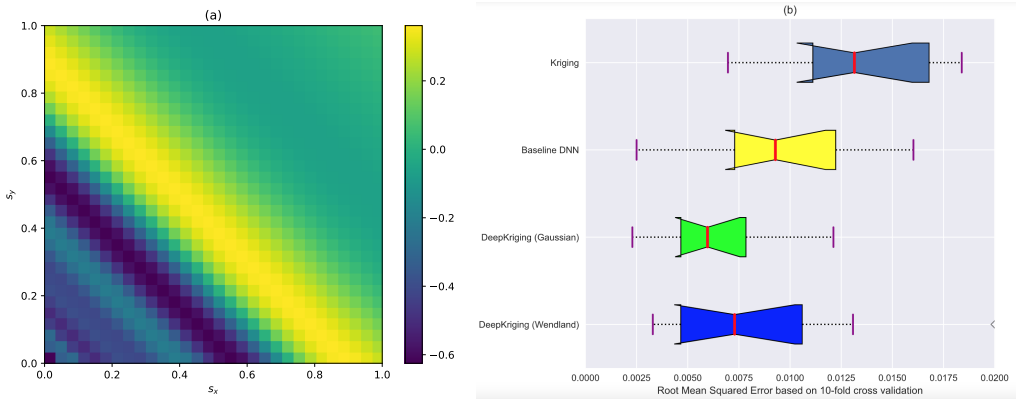

Figure S4: (a) Visualization of the simulated data generated from $Y(\mathbf{s}) = \sin\{30(\bar{s}-0.9)^4\}\cos\{2(\bar{s}-0.9)\} + (\bar{s}-0.9)/2$, where $\mathbf{s} = (s_x, s_y)^T \in [0,1]^2$ and $\bar{s} = (s_x + s_y)/2$. (b) Boxplots of the 10 RMSEs based on the 10-fold cross-validations from DeepKriging with Wendland kernel (blue), DeepKriging with Gaussian kernel (green), baseline DNN (yellow), and Kriging (gray).

Table S4: Model performance for both the training and testing sets based on 10-fold cross-validation. Data are generated from $Y(\mathbf{s}) = \sin\{30(\bar{s} - 0.9)^4\}\cos\{2(\bar{s} - 0.9)\} + (\bar{s} - 0.9)/2$, where $\mathbf{s} = (s_x, s_y)^T \in [0,1]^2$ and $\bar{s} = (s_x + s_y)/2$. RMSEs and MAPEs are computed, and the mean and standard deviations (SD) for the 10 sets of validation errors are given. DeepKriging I and II are the DeepKriging predictions with Wenland kernel and Gaussian kernel, respectively, based on the MSE loss. The baseline DNN includes only coordinates $\mathbf{s}$ in the features. The Kriging prediction uses an estimated exponential covariance function.

Models	DeepKriging I		DeepKriging II		Baseline DNN		Kriging	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Training set								
RMSE ($\times 10^{-3}$)	**.526**	.062	**1.382**	.082	6.946	3.946	1.757e-4	2.399e-4
MAPE	**.732**	.838	**1.386**	.622	9.634	6.389	5.152e-7	6.624e-8
Testing set								
RMSE ($\times 10^{-3}$)	**8.431**	4.927	**6.578**	3.065	9.987	5.375	15.135	7.804
MAPE	**10.520**	5.397	**5.798**	2.361	10.875	7.494	.098	.088

S4.2 DeepKriging is non-linear

It is important to investigate how the DeepKriging prediction is linked with the observations. We generate 100 observations generated by $z(s) = Y(s)\mathbb{1}_{\{Y(s)>0\}} + \varepsilon(s)$, where $Y(s) = 10\cos(20s)$, 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 S5(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, and obtain the prediction $\hat{Y}(s+1)$ by Kriging or DeepKriging. 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 S5(b) show that the Kriging prediction is linear in observations, while the DeepKriging provides an obviously nonlinear prediction in observations. This simulation study shows that although the Kriging and DeepKriging predictions are almost identical, the underlying relationships between the prediction and observations are totally different.

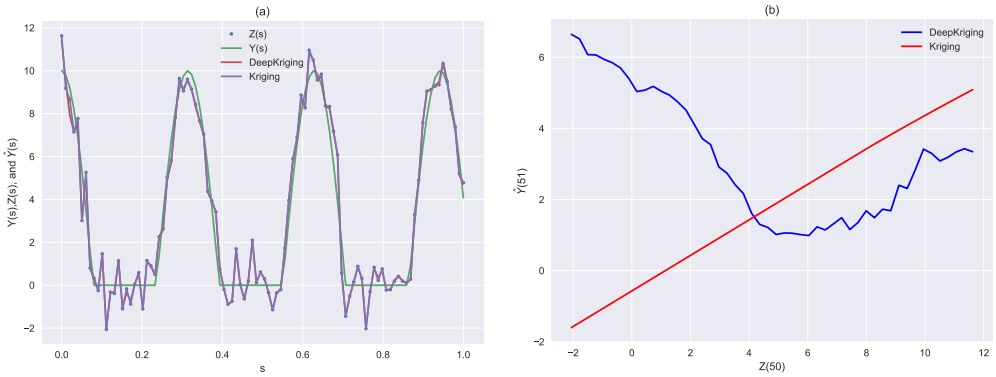

Figure S5: (a) Visualization of the simulated data and predictions. The observations (blue dots) z(s) are generated by the the signal (green line) $Y(s)\mathbb{1}_{\{Y(s)>0\}} + \varepsilon(s)$, where $Y(s) = 10\cos(20s)$, with a standard Gaussian noise. The predictions of Kriging (purple line) and DeepKriging (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.s 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.

S4.3 Computational time of DeepKriging

Based on the same simulation setting in Section S4.2, we investigate the computational time of DeepKriging compared to Kriging with different sample sizes N. Figure S6 shows the 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, but 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.

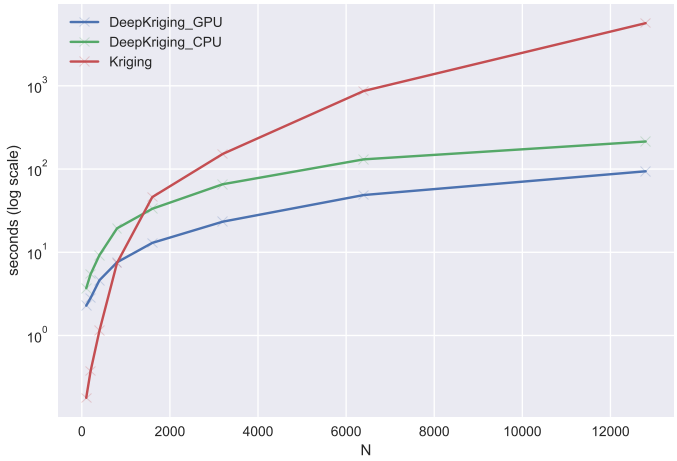

Figure S6: Computational time in seconds ($\log_{10}$ scale) of DeepKriging and Kriging for different sample sizes N. The green line is the computational time of Kriging. The blue and green lines are the computational time of DeepKriging based on a Tesla P100 GPU and a 2.5 GHz Intel Core i7 CPU, respectively.

S5. Distribution prediction and uncertainty quantification of DeepKriging

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 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.

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. Some studies try to combine deep learning and GPs which can be potentially applied in spatial prediction (Damianou and Lawrence, 2013; Lee et al., 2017; Kumar et al., 2018; Zammit-Mangion et al., 2019; Blomqvist et al., 2019). These methods are often called deep Gaussian processes that characterize deep learning in the framework of GPs via Bayesian learning (Neal, 1996). Based on the correspondence between an infinitely wide DeepKriging and GPs as illustrated in Section 3.3 in the main manuscript, we can potentially compute the covariance function for the induced GPs and then use the resulting GPs to perform Bayesian inference for wide DeepKriging networks. However, it is infeasible to train an infinitely wide DNN in practice, and Lee et al. (2017) found that the test performance increases as finite-width trained networks are made wider and

more similar to a GP, and thus that GP predictions typically outperform those of finite-width networks. The related computation burden is another issue for applications in large datasets.

The Bayesian methods mentioned above require configurations on the prior distributions for hyper-parameters. 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 using neural networks. Although the method was designed for a regression problem, we can extend the idea to our DeepKriging framework for predictive distribution estimation. The idea is straightforward and the method is easy to implement. We call the method deep distribution spatial prediction (DDSP), as illustrated below.

S5.1 Deep Distribution Spatial Regression

We quantify the uncertainty of 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 $f_{\text{UQ}}(y|x)$, whose support is $[l, u]$, with l and u being the lower and upper bound, respectively. 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:

$$\hat{f}_{\mathrm{UQ}}(y|x) = \sum_{m=1}^{M+1} \frac{\widehat{p}_m(\mathbf{s}_0)}{|T_m|} \mathbb{1}\{y \in T_m\}. \tag{S5.2}$$

The crucial step in Equation (S5.2) is to estimate the bin probabilities $\{\widehat{p}_1(\mathbf{s}_0), \ldots, \widehat{p}_{M+1}(\mathbf{s}_0)\}$ 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: $\sum_{n=1}^{N} \sum_{m=1}^{M+1} \mathbb{1}\{z(\mathbf{s}_n) \in T_m\} \log\{p_m(\mathbf{s}_n)\}$.

In practice, however, the estimation of a continuous density function by the multi-class fixed classification is sensitive to the choice of bins. Therefore, for DDSP in DeepKriging, we use the improved option with joint binary cross entropy loss function (JBCE) and ensembles. The JBCE function is specified as

$$\mathrm{JBCE} = \sum_{\mathrm{n}=1}^{\mathrm{N}} \sum_{\mathrm{m}=1}^{\mathrm{M}} \left[\mathbb{1}\{\mathrm{z}(\mathbf{s}_\mathrm{n}) \leq \mathrm{c_m}\} \log\{\mathrm{F}(\mathrm{c_m}; \mathbf{s}_\mathrm{n})\} + \mathbb{1}\{\mathrm{z}(\mathbf{s}_\mathrm{n}) > \mathrm{c_m}\} \log\{1 - \mathrm{F}(\mathrm{c_m}; \mathbf{s}_\mathrm{n})\}\right], \tag{S5.3}$$

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 pro-

vides 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 one of the M bins, where the $m-th$ bin is $T_{im} = [c_{i,m-1}, c_{im})$;
  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 the density regression provides an uncertainty quantification 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 ensembles, 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 pat-

tern 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.

S5.2 Uncertainty Quantification: A Simulation Study

The simulated data is generated by a 1-D 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 $[0, 1]$, out of which $2,000$ are training samples and 500 are testing samples.

Figure S7 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. Moreover, the prediction band of the Kriging is too narrow for large s and two large for small s.

Numerical evaluations on testing samples are further used for comparison based on the root mean squared error (RMSE) and average quantile loss (AQTL). AQTL is defined by $\mathrm{AQTL} = \sum_{\mathrm{t}=1}^{99} \sum_{\mathrm{n}=1}^{\mathrm{N}} (\{\mathrm{z}(\mathrm{s_n}) - \widehat{\mathrm{Q}}(\mathrm{t}/100)\}[\mathrm{t}/100 - \mathbb{1}\{\mathrm{z}(\mathrm{s_n}) \leq \widehat{\mathrm{Q}}(\mathrm{t}/100)\}])$, where $\widehat{Q}(t/100)$ is the estimated t-th percentile.

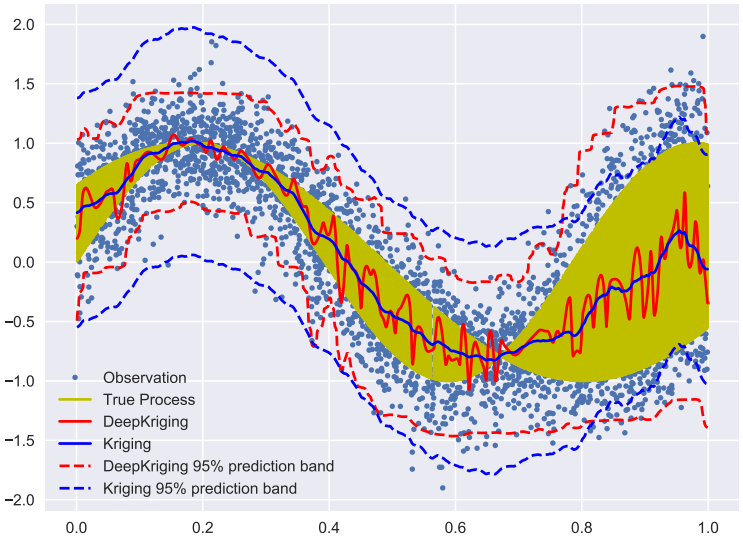

Figure S7: Prediction results for a Gaussian mixture model. The blue dots are the observations simulated from the Gaussian mixture model. The solid lines are the true process (yellow) and the predictions from DeepKriging (red) and Kriging (blue), respectively. The dashed lines represent the 95% prediction bands from DeepKriging (red) and Kriging (blue), respectively.

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 of DeepKriging (0.12), which further implies that Kriging is only a good estimation in terms of the mean squared error which reflects the mean and variance, but cannot reflect other important properties such as quantiles.

S6. 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 for this work are written in Python. We also provide 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

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.

Blomqvist, K., Kaski, S., and Heinonen, M. (2019). Deep convolutional gaussian processes. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 582–597. Springer.

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

Damianou, A. and Lawrence, N. (2013). Deep gaussian processes. In *Artificial Intelligence and Statistics*, pages 207–215.

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.

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.

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

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

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kumar, V., Singh, V., Srijith, P., and Damianou, A. (2018). Deep gaussian processes with convolutional kernels. *arXiv preprint arXiv:1806.01655*.

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*.

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

Neal, R. M. (1994). Priors for infinite networks. *Technical Report*.

Neal, R. M. (1996). *Bayesian Learning for Neural Networks*, volume 118. Springer Science & Business Media.

Nychka, D., Bandyopadhyay, S., Hammerling, D., Lindgren, F., and Sain, S. (2015). A multi-resolution 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*.

Zammit-Mangion, A., Ng, T. L. J., Vu, Q., and Filippone, M. (2019). Deep compositional spatial models. *arXiv preprint arXiv:1906.02840*.