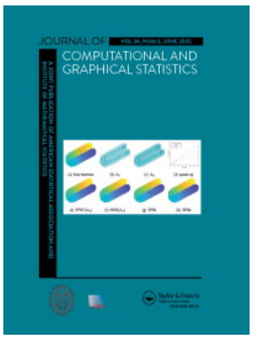

Taylor & Francis
Taylor & Francis Group

Journal of Computational and Graphical Statistics

ISSN: 1061-8600 (Print) 1537-2715 (Online) Journal homepage: www.tandfonline.com/journals/ucgs20

Efficient Large-Scale Nonstationary Spatial Covariance Function Estimation Using Convolutional Neural Networks

Pratik Nag, Yiping Hong, Sameh Abdulah, Ghulam A. Qadir, Marc G. Genton & Ying Sun

To cite this article: Pratik Nag, Yiping Hong, Sameh Abdulah, Ghulam A. Qadir, Marc G. Genton & Ying Sun (2025) Efficient Large-Scale Nonstationary Spatial Covariance Function Estimation Using Convolutional Neural Networks, Journal of Computational and Graphical Statistics, 34:2, 683-696, DOI: 10.1080/10618600.2024.2402277

To link to this article: https://doi.org/10.1080/10618600.2024.2402277

Published online: 12 Nov 2024.

JOURNAL OF COMPUTATIONAL AND GRAPHICAL STATISTICS
2025, VOL. 34, NO. 2, 683–696
https://doi.org/10.1080/10618600.2024.2402277

Efficient Large-Scale Nonstationary Spatial Covariance Function Estimation Using Convolutional Neural Networks

Pratik Nag[a], Yiping Hong[b], Sameh Abdulah[a], Ghulam A. Qadir[c], Marc G. Genton[a], and Ying Sun[a]

[a]Statistics Program, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia; [b]School of Mathematics and Statistics, Beijing Institute of Technology, Beijing, China; [c]Computational Statistics (CST) Group, Heidelberg Institute for Theoretical Studies (HITS), Heidelberg, Germany

ABSTRACT

Spatial processes observed in various fields, such as climate and environmental science, often occur at large-scale and demonstrate spatial nonstationarity. However, fitting a Gaussian process with a nonstationary Matérn covariance is challenging, as it requires handling the complexity and computational demands associated with modeling the varying spatial dependencies over large and heterogeneous domains. Previous studies in the literature have tackled this challenge by employing spatial partitioning techniques to estimate the parameters that vary spatially in the covariance function. The selection of partitions is an important consideration, but it is often subjective and lacks a data-driven approach. To address this issue, in this study, we use the power of Convolutional Neural Networks (ConvNets) to derive subregions from the nonstationary data by employing a selection mechanism to identify subregions that exhibit similar behavior to stationary fields. We rely on the `ExaGeoStat` software for large-scale geospatial modeling to implement the nonstationary Matérn covariance for large scale exact computation of nonstationary Gaussian likelihood. We also assess the performance of the proposed method with synthetic and real datasets at large-scale. The results revealed enhanced accuracy in parameter estimations when relying on ConvNet-based partition compared to traditional user-defined approaches.

ARTICLE HISTORY
Received June 2023
Accepted August 2024

KEYWORDS
Convolutional Neural Networks (ConvNets); Geospatial data; High-Performance Computing (HPC); Likelihood; Nonstationary Matérn covariance; Spatial domain partitions

1. Introduction

Gaussian processes (GPs) are fundamental statistical models used across diverse domains such as machine learning, statistics, signal processing, and physics. Within spatial statistics, GPs find extensive application in modeling, simulating, and predicting spatial and spatio-temporal data, notably through techniques like kriging. However, at large-scale, complex spatial processes often exhibit nonstationarity, influenced by factors such as spatial evolution within the underlying physical processes, heterogeneity in terrain between land and ocean, and variations in topographical elevation. One commonly employed strategy for tackling this nonstationarity involves using nonstationary covariance functions.

Risser (2016) reviewed various approaches to modeling covariance nonstationarity. The spatial deformation method is a well-known technique for modeling nonstationary spatial data (Sampson and Guttorp 1992; Anderes and Stein 2008; Qadir, Sun, and Kurtek 2021). It is often coupled with GPs to introduce nonstationarity into the model. In deformation-based nonstationary GPs, the estimation process involves finding an injective deformation function that transforms the spatial coordinates. This transformation aims to make the process stationary when viewed in the transformed space of coordinates, often referred to as the deformed space. While the deformation-based approach offers flexibility for modeling nonstationary spatial data, its success heavily relies on accurately specifying and estimating the deformation function, which can be challenging when dealing with a single realization of the process. Additionally, the deformation-based method can be computationally intensive, particularly for large datasets or high-dimensional problems. The differential operator approach by Jun and Stein (2008) and Lindgren, Rue, and Lindström (2011) is another alternative method for nonstationary spatial data modeling, which involves the use of differential operators to capture the spatial or temporal variation in the data. This approach allows for flexible modeling of nonstationarity and can capture a wide range of spatial or temporal patterns. Although it can be extended to big datasets, the computation can become intensive. Another well-established alternate method for nonstationary spatial modeling is the process convolution approach (Paciorek and Schervish 2006; Wilson and Adams 2013). This approach is based on the convolution of a stationary process with a nonstationary kernel to obtain a nonstationary process. However, the scalability of the process convolution method for large datasets remains a computationally challenging task.

Among the aforementioned nonstationary methods, the convolution-based approach is arguably a preferred choice among geostatisticians. This approach has resulted in a closed-form nonstationary Matérn covariance model, adding to its

CONTACT Ying Sun *ying.sun@kaust.edu.sa* Statistics Program, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia.

appeal and practicality in nonstationary spatial data modeling. The nonstationary Matérn covariance model allows for spatially varying covariance parameters that characterize different types of nonstationarity. Accordingly, the parameters can be expressed as a function of spatial coordinates that vary across space. This spatially varying specification of covariance parameters poses challenges in estimation, as the number of estimable parameters increases with the growing number of spatial locations. Accuracy and efficiency in such an estimation are challenging to achieve. To address this problem, many implementations assume that the covariance function is locally stationary (Paciorek and Schervish 2006; Anderes and Stein 2011) or weighted locally stationary (Fouedjio, Desassis, and Rivoirard 2016; Risser and Calder 2017; Li and Sun 2019). These assumptions are rooted in partitioning the spatial domain into distinct subregions, where the spatially varying covariance parameters are assumed to be either constant or exhibit slow changes within each subregion. This simplification greatly facilitates model fitting in numerous applications. However, as a result, the choice of region partitioning substantially impacts the performance of the model. The existing literature primarily relies on subjective partitioning or fixed splitting methods, thus lacking a data-driven approach.

In an alternative perspective, spatial data can be conceptualized as an incomplete image, wherein certain pixels are missing or deleted. In contrast to traditional images that use RGB values to describe each pixel, spatial data are represented by continuous values that convey specific information related to the spatial domain. As a result, spatial data analysis can be viewed as a specialized analog of image processing. In image processing, deep learning models that are composed of multiple layers of nonlinear processing have shown impressive results in feature extraction, transformation, pattern analysis, and classification. Convolutional Neural Networks (ConvNets), first introduced by LeCun et al. (1989), have emerged as the leading architecture for image recognition, classification, and detection tasks (LeCun, Bengio, and Hinton 2015; Zhang et al. 2016). Interestingly, ConvNets are increasingly making their presence felt in the field of spatial statistics, as evidenced by recent studies (Gerber and Nychka 2021; Lenzi et al. 2023; Lenzi and Rue 2023).

Consequently, we present a novel data-driven framework that leverages the image-processing capabilities of modern ConvNets to identify optimal subregions for nonstationary spatial data modeling. The optimal choice uses classifying spatial fields through ConvNets into stationary and nonstationary spatial fields. This framework is based on analyzing the nonstationarity index, which characterizes the prevalent nonstationarity observed in spatial data. Subsequently, optimal subregions are pinpointed, and parameters for each subregion are estimated via kernel smoothing. We employ the `ExaGeoStat` software to facilitate scalable parameter estimation, designed explicitly for modeling large-scale geospatial data (Abdulah et al. 2018a). Through this approach, we enhance the estimation of spatially varying parameters for nonstationary Matérn covariance functions, circumventing the need for approximated likelihoods and surpassing the limitations of traditional fixed-splitting approaches.

The article is structured as follows: Section 2 presents an introduction to the nonstationary Matérn covariance function and its implementation in `ExaGeoStat`. Section 3.1 discusses the proposed ConvNet algorithm, including its stationary and nonstationary classification implementation. Section 3.2 describes the data pre-processing required for reconstructing irregularly spaced data into a 100×100 gridded input for the ConvNet model. Section 3.4 discusses the algorithm of subregion selection through the ConvNet model. Section 4 presents the empirical evaluation of the proposed approach on synthetic datasets. Section 5 applies the proposed approach to soil moisture data. Conclusions and discussions are provided in Section 6.

2. Gaussian Likelihood Computation

2.1. Nonstationary Matérn Covariance Function

The univariate Gaussian random field (GRF), $\{Z(\boldsymbol{s}),\ \boldsymbol{s} \in D \subset \mathbb{R}^d\}$ with spatial location indexing $\boldsymbol{s}$, can be expressed as

$$Z(\boldsymbol{s}) = \mu(\boldsymbol{s}) + Y(\boldsymbol{s}) + \epsilon(\boldsymbol{s}), \quad \boldsymbol{s} \in D, \tag{1}$$

where $\mu(\cdot)$ is the mean function, $Y(\cdot)$ is a zero-mean GP with covariance function $C(\cdot,\cdot)$ and $\epsilon(\cdot) \sim \mathcal{N}(0, \tau^2(\cdot))$ is the nugget effect attributed to measurement inaccuracy and environmental variability. Here $\mu(\cdot)$ is assumed to be constant for simplicity, the covariance function $C(\cdot,\cdot) = C(\cdot,\cdot;\boldsymbol{\theta}_0)$ has a parametric form with associated parameters $\boldsymbol{\theta}_0 \in \mathbb{R}^p$, and the random processes $Y(\cdot)$ and $\epsilon(\cdot)$ are independent. The equation (1) provides a general representation of a GRF, accommodating both stationary and nonstationary processes.

Let $\boldsymbol{\theta} = (\boldsymbol{\theta}_0^\top, \tau)^\top$ represent a vector of unknown parameters. We consider $C(\cdot,\cdot) = C(\cdot,\cdot;\boldsymbol{\theta})$ to be an isotropic covariance function that incorporates the nugget term as an addition on the diagonal of the associated covariance matrix. Isotropic covariance functions exhibit rotational and reflection invariance, which is a restriction of the translation-invariant stationary covariance functions. The literature on spatial statistics offers a variety of isotropic covariance models, including the spherical, Matérn, and Cauchy models (Cressie 2015). The generalization of an isotropic covariance function into an anisotropic nonstationary covariance function is a complex and challenging task. The Matérn class achieved this nontrivial generalization through the process convolution approach (Higdon 1998; Stein 2005; Paciorek and Schervish 2003, 2006), and as a result, we have a closed-form nonstationary Matérn covariance function $C^{NS}(\cdot,\cdot;\boldsymbol{\theta}) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$:

$$\begin{aligned} C^{NS}\left(\boldsymbol{s}_i, \boldsymbol{s}_j; \boldsymbol{\theta}\right) &= \tau\left(\boldsymbol{s}_i\right) \tau\left(\boldsymbol{s}_j\right) \mathbb{1}_{ij}\left(\boldsymbol{s}_i, \boldsymbol{s}_j\right) \\ &+ \frac{\sigma\left(\boldsymbol{s}_i\right) \sigma\left(\boldsymbol{s}_j\right) \left|\boldsymbol{\Sigma}\left(\boldsymbol{s}_i\right)\right|^{1/4} \left|\boldsymbol{\Sigma}\left(\boldsymbol{s}_j\right)\right|^{1/4}}{\Gamma(\bar{\nu}(\boldsymbol{s}_i, \boldsymbol{s}_j)) 2^{\bar{\nu}(\boldsymbol{s}_i, \boldsymbol{s}_j)-1}} \left|\frac{\boldsymbol{\Sigma}\left(\boldsymbol{s}_i\right) + \boldsymbol{\Sigma}\left(\boldsymbol{s}_j\right)}{2}\right|^{-1/2} \\ &\times \left(2\sqrt{\bar{\nu}(\boldsymbol{s}_i, \boldsymbol{s}_j) Q_{ij}}\right)^{\bar{\nu}(\boldsymbol{s}_i, \boldsymbol{s}_j)} \mathcal{K}_{\bar{\nu}(\boldsymbol{s}_i, \boldsymbol{s}_j)}\left(2\sqrt{\bar{\nu}(\boldsymbol{s}_i, \boldsymbol{s}_j) Q_{ij}}\right). \end{aligned} \tag{2}$$

Here, $\bar{\nu}(\boldsymbol{s}_i, \boldsymbol{s}_j) = \frac{\nu(\boldsymbol{s}_i)+\nu(\boldsymbol{s}_j)}{2}$, $\sigma(\cdot) : \mathbb{R}^d \to \mathbb{R}_+$ is the spatially varying standard deviation, $\tau(\cdot) : \mathbb{R}^d \to \mathbb{R}_+$ is the spatially varying nugget, $\nu(\cdot) : \mathbb{R}^d \to \mathbb{R}_+$ is the spatially varying smoothness, $\boldsymbol{\Sigma}(\cdot) : \mathbb{R}^d \to \mathbb{R}^{d\times d}$ is the spatially varying anisotropy, and Q_{ij} is the Mahalanobis distance between $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$ such that $Q_{ij} = (\boldsymbol{s}_i - \boldsymbol{s}_j)^\top \left(\frac{\boldsymbol{\Sigma}(\boldsymbol{s}_i)+\boldsymbol{\Sigma}(\boldsymbol{s}_j)}{2}\right)^{-1} (\boldsymbol{s}_i - \boldsymbol{s}_j)$, and $\mathcal{K}_\nu(\cdot) : \mathbb{R}_+ \to \mathbb{R}$

is the modified Bessel function of the second kind with order $\nu > 0$. Specifically, for $d = 2$, the spatially varying anisotropy can be decomposed as

$$\boldsymbol{\Sigma}(\boldsymbol{s}_i) = \begin{bmatrix} \cos(\phi(\boldsymbol{s}_i)) & -\sin(\phi(\boldsymbol{s}_i)) \\ \sin(\phi(\boldsymbol{s}_i)) & \cos(\phi(\boldsymbol{s}_i)) \end{bmatrix} \begin{bmatrix} \lambda_1(\boldsymbol{s}_i) & 0 \\ 0 & \lambda_2(\boldsymbol{s}_i) \end{bmatrix} \times \begin{bmatrix} \cos(\phi(\boldsymbol{s}_i)) & -\sin(\phi(\boldsymbol{s}_i)) \\ \sin(\phi(\boldsymbol{s}_i)) & \cos(\phi(\boldsymbol{s}_i)) \end{bmatrix},$$

where $\phi(\cdot) : \mathbb{R}^d \to (0, \pi/2]$ represents the angle of rotation and $\lambda_1(\cdot) : \mathbb{R}^d \to \mathbb{R}_+$, $\lambda_2(\cdot) : \mathbb{R}^d \to \mathbb{R}_+$ are the eigenvalues that represent the spatially varying correlation range. Note that all the covariance parameters in (2) are indeed a function of spatial locations, and therefore, the parameter vector $\boldsymbol{\theta}$ can be redefined as $\boldsymbol{\theta}(\boldsymbol{s}_i) = \{\sigma(\boldsymbol{s}_i), \lambda_1(\boldsymbol{s}_i), \lambda_2(\boldsymbol{s}_i), \phi(\boldsymbol{s}_i), \tau(\boldsymbol{s}_i), \nu(\boldsymbol{s}_i)\}^\top$. Theoretically, all these parameters can be allowed to vary spatially, which would lead to many rich classes of nonstationarity. However, in favor of the optimization feasibility and detectability issue, certain reasonable restrictions on the spatially varying parameters are considered in practice (Anderes and Stein 2008).

2.2. Maximum Likelihood Parameter Estimation with ExaGeoStat

Maximum likelihood estimation (MLE) is a commonly employed statistical method for estimating the model parameters of the underlying field. In the context of Gaussian processes (GPs), MLE aims to maximize the Gaussian likelihood function over the parameter space for a given dataset (MacKay 1998). Abdulah et al. (2018a) have developed the `ExaGeoStat` software to estimate the covariance parameters over a given geospatial domain in parallel and at large-scale. `ExaGeoStat` has distributed memory support, enabling one to perform parallel computing. It also provides a version that supports the same functionalities in R environment (Abdulah et al. 2023). As a result, large datasets can be incorporated and analyzed effectively through this software on shared-memory systems and distributed memory systems with and without GPU accelerators. `ExaGeoStat` also supports the simulation of stationary Gaussian processes (GPs). In this work, we implement the nonstationary Matérn covariance in `ExaGeoStat` for large-scale exact simulation of nonstationary random fields along with its parameter estimation, which was previously unavailable in the literature. We also selected the kernel smoothing method for the parameter estimation, which will be discussed in the next section.

2.3. Kernel Smoothing

For parameter estimation, the likelihood function is maximized for the set of parameter vectors, which in the case of the nonstationary covariance function C^{NS} is the parameter set $\boldsymbol{\theta}$. Here, we assume the mean function is constant or zero. However, if a more general mean function is considered, a restricted maximum likelihood method may be necessary (Patterson 1975). The MLE method for large datasets can be challenging due to the computationally intensive nature of large matrix operations. Additionally, when dealing with spatially varying parameters, the task becomes even more difficult due to many unknown parameters that should be estimated.

Numerous studies have tackled this challenge through quicker likelihood approximation methods (Sun, Li, and Genton 2012), such as spectral approximations (Whittle 1954) and composite likelihoods (Vecchia 1988), which make the optimization computationally feasible on traditional machines. Other approaches rely on advanced parallel linear algebra libraries to support large-scale MLE (Abdulah et al. 2018a, 2018b). To tackle the issue of many unknown parameters, we use a kernel smoothing approach to represent the spatially varying parameters through smooth functions. This representation has been extensively used in spatial statistics to estimate spatially varying parameters, including the covariance function in Gaussian random fields (Higdon 1998; Paciorek and Schervish 2003; Stein 2005; Li and Sun 2019). It uses a discrete mixture of the parameters at representative locations called anchor locations through kernel convolution. Specifically, the spatially varying parameters at any given location $\boldsymbol{s}_i$ is

$$\boldsymbol{\theta}(\boldsymbol{s}_i) = \sum_{k=1}^{K} W(\boldsymbol{s}_i, \boldsymbol{S}_k)\, \boldsymbol{\theta}(\boldsymbol{S}_k), \tag{3}$$

where $\{\boldsymbol{S}_1, \ldots, \boldsymbol{S}_K\}$ are the representative locations at K such subregions, $\boldsymbol{\theta}(\boldsymbol{S}_k) \equiv \{\sigma(\boldsymbol{S}_k), \lambda_1(\boldsymbol{S}_k), \lambda_2(\boldsymbol{S}_k), \phi(\boldsymbol{S}_k), \tau(\boldsymbol{S}_k), \nu(\boldsymbol{S}_k)\}^\top$ is the parameter vector at the representative location $\boldsymbol{S}_k, k = 1, \ldots, K$, the weights are $W(\boldsymbol{s}_i, \boldsymbol{S}_k) = \frac{\mathcal{K}^*(\boldsymbol{s}_i, \boldsymbol{S}_k)}{\sum_{k=1}^{K} \mathcal{K}^*(\boldsymbol{s}_i, \boldsymbol{S}_k)}$, and the kernel $\mathcal{K}^*(\boldsymbol{s}_i, \boldsymbol{S}_k) = \exp\left(-\|\boldsymbol{s}_i - \boldsymbol{S}_k\|^2/(2h)\right)$ is a Gaussian kernel with bandwidth h. The representative or anchor locations can simply be chosen to be the center of each of the sub-regions.

We estimate the parameters by implementing this kernel smoothing approach in `ExaGeoStat`. For our implementation, we have chosen three spatially varying parameters $\{\sigma(\boldsymbol{s}_i), \lambda(\boldsymbol{s}_i), \nu(\boldsymbol{s}_i)\}$ with $\lambda_1(\mathbf{s}) = \lambda_2(\mathbf{s}) = \lambda(\mathbf{s})$ and ϕ, τ to be constant. The above representation of the spatially varying parameters can capture various spatially varying functions for the covariance parameters. However, the accuracy of the functional form of the spatially varying parameters heavily depends on the choice of the subregions. This article proposes a ConvNet-based subregion selection algorithm to provide better parameter estimates than the existing subjective choices.

3. ConvNet for Nonstationarity Classification

Neural Networks (NN) are a class of machine learning models that consist of interconnected nodes, or neurons, organized into layers that process input data and produce output predictions. Through training on large datasets, neural networks can learn complex patterns and relationships, making them powerful tools for tasks such as image recognition, natural language processing, and predictive modeling. Convolutional Neural Networks (ConvNets) are a specific type of NN that can effectively process and analyze data with a grid-like structure, such as images or time series data (Rawat and Wang 2017; Zhiqiang and Jun 2017; Ajmal et al. 2018). Unlike traditional NNs that process data only in fully connected sequential layers, ConvNets additionally employ convolutional layers that apply filters or kernels to small,

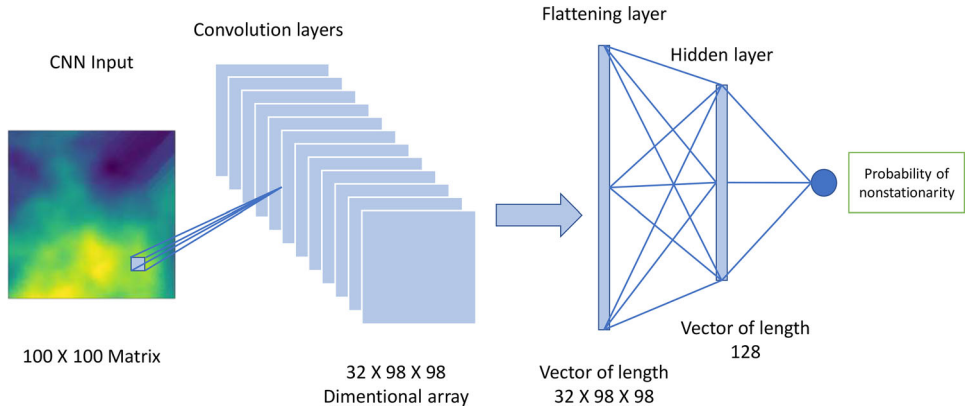

Figure 1. Visualization of the architecture of the proposed ConvNet model.

overlapping input regions. These filters detect many image features, such as edges, textures, or shapes, by convolving across the input data. This convolutional operation allows the network to capture local dependencies and preserve spatial relationships within the data. This study uses ConvNets to classify stationary and nonstationary random fields for spatial data.

3.1. The ConvNet Architecture

Our proposed ConvNet model effectively distinguishes between stationary and nonstationary regions by taking irregular spatial locations as input. The model generates a probability score indicating the likelihood of a region being stationary. Regions are classified as nonstationary if the probability equals or exceeds 0.5. Conversely, if the probability is below 0.5, the region is categorized as stationary. Figure 1 visually illustrates the components of our ConvNet model, which includes the following components:

- **Convolution layer**: In this layer, the preprocessed data are fed into a ConvNet through the input layer of dimension 100×100, which captures the spatial neighborhood structure. The input data are derived from a regular grid of size 100×100, and the convolutional layer uses 3×3 kernel. As a result, this layer produces 32 feature maps, each consisting of 98×98 values in its output. More specifically, let $[\tilde{Z}(i,j)]_{i,j=1}^{100}$ represent the input to this layer. The output of the layer is denoted as $[m_{i,j}^{(\eta)}]_{i,j=1}^{98}$, where $\eta \in \{1,\ldots,32\}$ corresponds to each feature map. The value of $m_{i,j}^{(\eta)}$ is determined by the equation:
$$m_{i,j}^{(\eta)} = \tilde{\sigma}\left(\sum_{r=1}^{3}\sum_{s=1}^{3}\tilde{Z}(i+r-1,j+s-1)k_{r,s}^{(\eta)} + b^{(\eta)}\right),$$
where $\tilde{\sigma}(x) = \max(0,x)$ represents the Rectified Linear Unit (RELU) activation function. The kernel matrix $[k_{r,s}^{(\eta)}]_{r,s=1}^{3} \in \mathbb{R}^{3\times3}$ and the bias $b^{(\eta)} \in \mathbb{R}$, $\eta = 1,\ldots,32$, are specific to this layer, resulting in a total of $(3\times3+1)\times32 = 320$ parameters.
- **Flatten layer**: The role of this layer is to convert the outputs from the preceding convolutional layer into a flattened vector of length $32 \times 98 \times 98$. Specifically, the tensor of dimension $32 \times 98 \times 98$ is concatenated into a single vector of length $N_F = 32 \times 98 \times 98 = 307,328$.
- **Fully connected layer**: In this layer, the flattened vector from the previous layer undergoes two sequential nonlinear transformations. Firstly, the incoming vector of length N_F is transformed into a hidden layer vector of length $H_1 = 128$ through an elementwise RELU activation operation on the affine transformation with weights and biases. Similarly, it is further transformed into a vector of length $H_2 = 2$. The vector input to Hidden Layer 1 is denoted as $[\tilde{m}_i]_{i=1}^{N_F}$, and the output is represented by $[\tilde{m}_i^{(1)}]_{i=1}^{H_1}$. The computation of $[\tilde{m}_i^{(1)}]_{i=1}^{H_1}$ is given as follows:
$$\tilde{m}_i^{(1)} = \tilde{\sigma}\left(\sum_{j=1}^{N_F} p_{i,j}^{(1)}\tilde{m}_j + p_{0,j}^{(1)}\right),$$
where $[p_{i,j}^{(1)}]_{1\le i\le H_1, 1\le j\le N_F} \in \mathbb{R}^{H_1\times N_F}$ and $[p_{0,i}^{(1)}]_{i=1}^{H_1}$ represent the parameters in terms of weights and biases, respectively. The total number of parameters in this layer is $(N_F+1)\times H_1$.

  Moving up to Hidden Layer 2, the input is $[\tilde{m}_i^{(1)}]_{i=1}^{H_1}$, and the output is $[\tilde{m}_i^{(2)}]_{i=1}^{H_2}$, where the computation of the outputs is given by:
$$\tilde{m}_i^{(2)} = \tilde{\sigma}\left(\sum_{j=1}^{H_1} p_{i,j}^{(2)}\tilde{m}_j^{(1)} + p_{0,j}^{(2)}\right),$$
where $[p_{i,j}^{(2)}]_{1\le i\le H_2, 1\le j\le H_1} \in \mathbb{R}^{H_2\times H_1}$ and $[p_{0,i}^{(1)}]_{i=1}^{H_2}$ represent the parameters in terms of weights and biases, respectively. The total number of parameters in this layer is $(H_1+1)\times H_2$.
- **Output layer**: The purpose of this layer is to generate the nonstationarity index by transforming the vector of length 2. The inputs to this layer are $\tilde{m}_1^{(2)}$ and $\tilde{m}_2^{(2)}$, and the output is obtained through the following expression:
$$I_{nonstat} = \frac{\exp(\tilde{m}_1^{(2)})}{\exp(\tilde{m}_1^{(2)}) + \exp(\tilde{m}_2^{(2)})},$$
which is the probability of the random field being nonstationary.

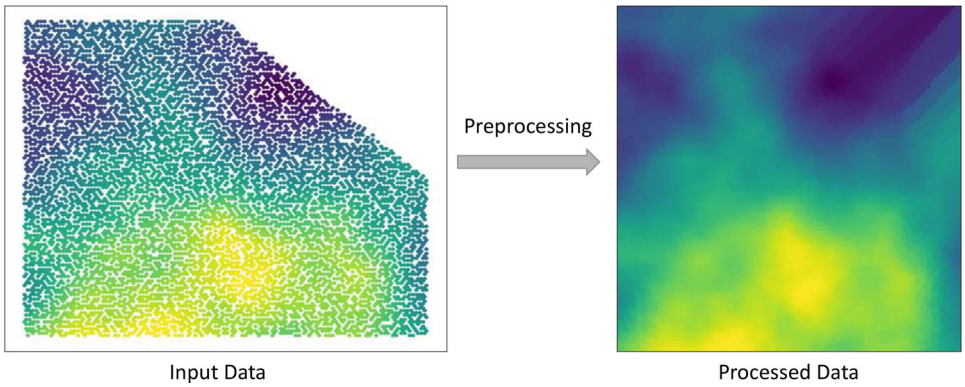

Figure 2. Data pre-processing step.

3.2. The ConvNet Data Preprocessing

Training the ConvNet discussed in Section 3.1 can be directly accomplished with regularly gridded datasets. However, real-world applications often involve irregularly spaced spatial data. To address this, we have developed a data preprocessing stage. Figure 2 illustrates the idea behind the pre-processing step. Our goal is to obtain a grid-like approximation for the irregularly spaced data. To achieve this, we use Splitting, Averaging, and Scaling operations. It is important to note that ideally, we aim for the raw data to closely resemble regularly gridded data. However, the further the raw data deviates from this grid-like structure, the more extensive the operations required to find a suitable grid-like proxy.

To pre-process the spatial field $Z(\boldsymbol{s})$ observed at n locations, recall the raw data as $\boldsymbol{Z} = (Z(\boldsymbol{s}_1), \ldots, Z(\boldsymbol{s}_n))^\top$, where $\boldsymbol{s}_i \in D$. Our goal is to map the locations from D to $[0,1]^2$, assuming that the observed region can be properly stretched. The resulting pre-processed data are $\tilde{\boldsymbol{Z}} = [\tilde{Z}(i,j)]_{i,j=1}^{100}$, which represent spatial data observed on a 100×100 regular grid. The steps can be broken down into the following:

- **Splitting:** We divide the observed region $[0,1]^2$ to 100×100 subregions indexed by (i,j) equally. Here the subregions $D_{i,j}$ are defined by

$$D_{i,j} = \begin{cases} [(i-1)/100, i/100) \times [(j-1)/100, j/100), & i,j < 100; \\ [99/100, 1] \times [(j-1)/100, j/100), & i = 100, j < 100; \\ [(i-1)/100, i/100) \times [99/100, 1], & i < 100, j = 100; \\ [99/100, 1] \times [99/100, 1], & i,j = 100. \end{cases}$$

- **Averaging:** Let $N_{i,j}$ be the total number of observations within $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n$ that lie within the subregion $D_{i,j}$. We define $\bar{Z}(i,j)$ as the mean of the observations within $D_{i,j}$. More specifically, we define

$$\bar{Z}(i,j) = \begin{cases} \frac{1}{N_{i,j}} \sum_{k:\boldsymbol{s}_k \in D_{i,j}} Z(\boldsymbol{s}_k), & N_{i,j} > 0; \\ \bar{Z}(u,v), \text{ where } (u,v) = \operatorname{argmin}_{u,v} & \\ \quad \{(i-u)^2 + (j-v)^2 : N_{u,v} > 0\}, & N_{i,j} = 0. \end{cases}$$

- **Scaling:** The averaging results $\bar{Z}(i,j)$ are scaled such that the range of the observations is $[0,1]$. Let $m = \min_{i,j \in 1,\ldots,100} \bar{Z}(i,j)$, $M = \max_{i,j \in 1,\ldots,100} \bar{Z}(i,j)$. We have

$$\tilde{Z}(i,j) = \begin{cases} (\bar{Z}(i,j) - m)/(M-m), & m \neq M; \\ 0.5, & m = M. \end{cases}$$

3.3. The ConvNet Training Phase

To train the ConvNet model for classifying stationary and nonstationary data, we used the `TensorFlow` library. We employed the categorical cross-entropy loss function and the Adam optimizer throughout the training process. The model underwent training for 25 epochs. Post-training, the ConvNet model yields a nonstationarity probability for each spatial dataset. Our ConvNet model is designed as a standalone approach, enabling its utilization for classifying any stationary or nonstationary spatial field. Additional details and the implementation of the Python code can be found at *https://github.com/kaust-es/Spatial_classifier_DL.git*.

A combination of stationary and nonstationary synthetic datasets was employed to train the ConvNet. Stationary datasets were generated using the stationary Matérn covariance, following a similar approach as in Abdulah et al. (2018a). Specifically, the stationary datasets were generated from a stationary isotropic Gaussian random field with Matérn covariance given by:

$$\mathcal{M}(h;\boldsymbol{\theta}) = \frac{\sigma^2}{2^{\nu-1}\Gamma(\nu)} \left(\frac{h}{\alpha}\right)^{\nu} \mathcal{K}_{\nu}\left(\frac{h}{\alpha}\right), \tag{4}$$

where $\boldsymbol{\theta} = (\sigma^2, \alpha, \nu)^\top > \boldsymbol{0}$. The effective range h_{eff} is defined as the value such that $\mathcal{M}(h_{\text{eff}};\boldsymbol{\theta})/\sigma^2 = 0.05$. We set $\sigma^2 = 1$, $\nu = 1/8 + (3/16)k$ for $k \in \{0, 15\}$, and α was chosen such that $h_{\text{eff}} = 0.05 + 0.003k$ for $k \in \{0, \ldots, 999\}$. One sample was generated for each parameter combination, resulting in 16,000 stationary data samples.

We employed the nonstationary Matérn covariance defined in (2) for nonstationarity. We used three spatially varying parameters $\boldsymbol{\theta}(\mathbf{s}) = \{\sigma(\mathbf{s}), \lambda(\mathbf{s}), \nu(\mathbf{s})\}$ with the kernel. The generation of the spatially varying parameters was performed through two settings.

The first setting employed nonlinear functions to generate $\boldsymbol{\theta}(\mathbf{s})$. Letting $\boldsymbol{s} = (s_1, s_2)$, details of the nonstationary patterns are provided as follows:

-

$$\sigma^{(1)}(\boldsymbol{s}) = \frac{1}{2} \sin\left[r_A\{(s_1 - 0.5)\cos(\theta_A) + (s_2 - 0.5)\sin(\theta_A)\}\right] + \frac{1}{2},$$

where $r_A \in \{5, 10, 15, \ldots, 50\}$, $\theta_A \in \{k\pi/12 : k = 0, 1, \ldots, 11\} \cup \{\pi/4 + k\pi/2, k = 0, 1, 2, 3\}$. In random field $Z_S(\boldsymbol{s})$, we choose $\sigma^2 = 1$, $\nu \in \{0.5, 1, 1.5, 2\}$, and $h_{\text{eff}} \in \{0.1, 0.2, 0.4, 0.8, 1.6\}$.

- $$\sigma^{(2)}(\boldsymbol{s}) = \frac{(s_1 - 0.5)\cos(\theta_D) + (s_2 - 0.5)\sin(\theta_D)}{|\cos(\theta_D)| + |\sin(\theta_D)|} + \frac{1}{2},$$

  where $\theta_D \in \{k\pi/12 : k = 0, 1, \ldots, 11\} \cup \{\pi/4 + k\pi/2, k = 0, 1, 2, 3\}$. In random field $Z_S(\boldsymbol{s})$, we choose $\sigma^2 = 1, \nu \in \{k/4 : k = 1, \ldots, 8\}$, and $h_{\text{eff}} \in \{0.1 + k/16 : k = 0, \ldots, 24\}$.
- We first define

  $$\begin{bmatrix} s_1^\star \\ s_2^\star \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} + \begin{bmatrix} \cos\theta_E & -\sin\theta_E \\ \sin\theta_E & \cos\theta_E \end{bmatrix} \begin{bmatrix} s_1 - 0.5 \\ s_2 - 0.5 \end{bmatrix}$$

  and $\bar{s}^\star = (s_1^\star + s_2^\star)/2$. Let

  $$\sigma_0^{(3)}(\boldsymbol{s}) = 3\sin\{20(\bar{s}^\star + 1.9)\}\cos\{20(\bar{s}^\star - 1.2)^6\} + 0.6e^{\sin(25s_1^\star) + \sin(13s_2^\star)} + \frac{\bar{s}^\star - 0.2}{2},$$

  and $\sigma^{(3)}(\boldsymbol{s})$ is obtained by scaling $\sigma_0^{(3)}(\boldsymbol{s})$ such that the minimum and maximum of $\sigma^{(3)}(\boldsymbol{s})$ are 0 and 1 for all observation locations, respectively.

The functions we used for $\nu(\mathbf{s})$ are the following:

- $$\nu^{(1)}(\boldsymbol{s}) = \frac{1}{2}\sin\left[[r_B\{(s_1 - 0.5)\cos(\theta_B) + (s_2 - 0.5)\sin(\theta_B)\}]^{2p_B}\right] + \frac{1}{2},$$

  where $p_B \in \{1, 2, 3\}$, $r_B \in \{3, 4, \ldots, 10\}$, or $p_B = 4$, $r_B = 3$; $\theta_B \in \{k\pi/12 : k = 0, 1, \ldots, 11\} \cup \{\pi/4 + k\pi/2, k = 0, 1, 2, 3\}$. In random field $Z_S(\boldsymbol{s})$, we choose $\sigma^2 = 1$, $\nu \in \{0.5, 1\}$, and $h_{\text{eff}} \in \{0.2, 0.4, 0.8, 1.6\}$.
- $$\nu^{(2)}(\boldsymbol{s}) = \frac{\exp\left[\sin\{r_C(s_1 - 0.5)\cos(\theta_C)\} + \sin\{r_C(s_2 - 0.5)\sin(\theta_C)\}\right] - e^{-2}}{e^2 - e^{-2}},$$

  where r_C, θ_C are similar to the case of $u = 1$.

The functions we used for $\lambda(\mathbf{s})$ are as follows:

- $$\lambda(\mathbf{s}) = r_D \times e^{\sin(\frac{\pi}{2}s_1) + \sin(\frac{\pi}{2}s_2)}, \tag{5}$$

  where $r_D \in (0.01, 0.2)$.

The second setting mainly relies on generating spatially varying parameters through kernel convolution, as defined in Section 2.3. Here, we choose $K = \{2, 3, 4, 5, 6, 7, 8\}$, $\sigma(\mathbf{S}_k) \in (0.4, 2.4)$, $\nu(\mathbf{S}_k) \in (0.4, 3.4)$, $\lambda(\mathbf{S}_k) \in (0.01, 0.4)$, $k = 1, \ldots, K$ and follow (3) to obtain the spatially varying parameters.

The synthetic data were generated to encompass various nonstationary patterns commonly encountered in spatial domains. The inclusion of such a variety of types of parameter configurations allowed for a comprehensive training regimen, enabling ConvNet to learn discriminative features across different nonstationarity profiles.

Considerations regarding training data may vary by application domain due to the unique characteristics and underlying processes inherent in each domain. For instance, spatial phenomena in environmental monitoring may exhibit distinct nonstationarity patterns compared to economic indicators or medical imaging data. While establishing universal guidelines for training data selection may be challenging, our methodology allows for domain-specific adjustments to accommodate variations in nonstationarity profiles. Our primary aim in this article is to provide a robust method capable of distinguishing stationary and nonstationary Matérn covariances, as we intend to use this for subregion partitioning. Future research could explore the development of adaptable training frameworks informed by domain-specific knowledge and empirical insights.

Sensitivity analyses were conducted to evaluate the robustness of the classifier to different choices of training data. We chose test scenarios that were different from the training scenarios. Our results demonstrate a satisfactory level of robustness across various scenarios, indicating the potential for the proposed method to generalize to diverse spatial domains. Detailed studies are presented in Section 4.2. While no single classifier may be universally applicable, our findings suggest the feasibility of developing adaptable models capable of addressing a wide range of nonstationarity profiles. Further investigations into the generalizability of the method across different application domains may shed light on the development of more robust and versatile classifiers. It is important to acknowledge that achieving a successful classifier for a specific scenario may require substantial training data. Without an adequate training dataset, the model may lack the necessary generalization and could potentially overfit on testing datasets.

3.4. ConvNet for Subregion Selection

In this section, we introduce a data-driven subregion partition method via ConvNet. The proposed algorithm requires a predetermined number of subregions denoted by K. For a randomly selected K points, $\mathbf{A} = \{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_K\}$ from $\mathbf{S}_{vec} = \{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n\}$, we obtain the vector of distances $\mathbf{E} = \{\|\boldsymbol{s}_i - \boldsymbol{a}_k\|^2 : \boldsymbol{a}_k \in \mathbf{A}\}, \boldsymbol{s}_i \in \mathbf{S}_{vec}$. We then assign $(\boldsymbol{s}_i, Z(\boldsymbol{s}_i))$ to subregion $\mathcal{R}_k$, if $\mathbf{E}_k$ is minimum in $\mathbf{E}$. Next, we preprocess the data for each subregion and use ConvNet to determine the probability of being nonstationary. The sum of such probabilities for all subregions, denoted by $P = \sum_{k=1}^{K} p_k$, indicates the degree of nonstationarity. We repeat this whole procedure *iters* number of times and select the set of subregions with the lowest P as the final partition for estimating the nonstationary covariance function. The anchor locations $\mathbf{S}$ are then defined to be the center points of those subregions. Algorithm 1 summarizes the whole procedure.

The algorithm has been implemented within the `ExaGeoStat` software using C++. To import the ConvNet model trained in `Tensorflow` into `ExaGeoStat`, we used the `Tensorflow_C_API`, which is an open-source software for integrating Python models into C/C++. Additionally, we seamlessly integrated the C++ code into the C environment of `ExaGeoStat` using the `CppFlow` package. During the

Algorithm 1 ConvNet Subregion Selection Algorithm

Require: A set of locations $\mathbf{S}_{vec}$, and observations $\mathbf{Z} = \{Z(\mathbf{s}_1), \ldots, Z(\mathbf{s}_n)\}$, number of subregions K, and number of iterations *iters*
Ensure: Anchor locations $\mathbf{S} = \{\mathbf{S}_1, \ldots, \mathbf{S}_K\}$ for K subregions
  for $u = 1 : iters$ **do**
    Select K random points $\mathbf{A}$ from the given set of locations $\mathbf{S}_{vec}$
    for $i = 1 : n$ **do**
      Calculate $\mathbf{E}$ for $\boldsymbol{s}_i$
      Assign $(\boldsymbol{s}_i, Z(\boldsymbol{s}_i))$ to $\mathcal{R}_k^u$ if $\mathbf{E}_k$ is minimum
    end for
    ▷ K subregions $\{\mathcal{R}_1^u, \ldots, \mathcal{R}_K^u\}$ for uth iteration
    for $k = 1 : K$ **do**
      $\tilde{\mathbf{Z}}_k = PreProcess(\mathcal{R}_k^u)$ ▷ Follow (3.2)
      $p_k = \text{ConvNet}(\tilde{\mathbf{Z}}_k)$ ▷ Nonstationary probability of each region k
    end for
    $P = \sum_{k=1}^{K} p_k$
    if $min > P$ **then**
      $min = P$ and $index = u$
      $\{\mathcal{R}_1, \ldots, \mathcal{R}_K\} = \{\mathcal{R}_1^u, \ldots, \mathcal{R}_K^u\}$
    end if
  end for
  $\mathbf{S} = \{(m(\mathcal{R}_1), \ldots, m(\mathcal{R}_K)\}$ ▷ $m(\mathcal{R}_k)$ is the centre of the subregion $\mathcal{R}_k$

installation of the package, the ConvNet model is automatically downloaded and placed in the appropriate folder to enable the execution of the algorithm. By leveraging this algorithm, `ExaGeoStat` can obtain subregions and provide parameter estimates and predictions for the nonstationary covariance.

4. Simulation Studies

This section outlines a set of experiments aimed at achieving two main objectives. First, we evaluate the effectiveness of our proposed ConvNet model in classifying a given spatial region as either stationary or nonstationary, along with providing an associated probability value. This is assessed in two different simulation settings. Second, we assess the performance of the proposed ConvNet-based nonstationary parameter estimation method and compare its quality to the fixed-partitioning methods. To conduct large-scale modeling experiments, we rely on the `ExaGeoStat` framework.

4.1. Settings of the Synthetic Spatial Datasets

4.1.1. First Simulation Setting

To validate the effectiveness of our proposed ConvNet model in classifying spatial data regions, we use the spatial data generation tool within the `ExaGeoStat` framework to generate a set of synthetic training and testing datasets by following the spatially varying parameter generations as discussed in Section 3.3. The locations within the datasets were generated using the formula $n^{-1/2}(i - 0.5 + X_{i,j}, j - 0.5 + Y_{i,j})$, where $i, j \in 1, \ldots, n^{1/2}$ and the values of $X_{i,j}$ and $Y_{i,j}$ were generated from iid samples of the uniform distribution Unif$(-0.4, 0.4)$. This process ensured that the spatial locations of the data followed a specific pattern for analysis. We generated 16,000 stationary datasets and 16,000 nonstationary datasets using the isotropic stationary Matérn for stationary fields, employing the nonstationary Matérn with spatially varying parameters as detailed in Section 3.3. We randomly selected 12,800 stationary datasets and 12,800 nonstationary datasets for training the ConvNet model, while the remaining datasets were reserved for testing purposes. The generated synthetic datasets underwent preprocessing before the training and testing phases, following the method outlined in Section 3.2. As a result, the data meet the necessary criteria for our ConvNet model. To train the ConvNet model, we preprocessed each dataset to fit within a regular square grid of size 100×100.

4.1.2. Second Simulation Setting

To assess the generalization of our model, we established an alternative simulation scenario. Here, we generated the training datasets using both isotropic stationary Matérn, as defined in (4), and nonstationary Matérn with spatially varying parameters, represented as

$$\begin{aligned}\lambda(\mathbf{s}) &= 0.04 \cdot \exp(\sin(0.5\pi s_1) + \sin(0.5\pi s_2)),\\ \sigma(\mathbf{s}) &= (0.33 \cdot \exp(-(s_1 + s_2))) + 0.8,\\ \nu(\mathbf{s}) &= 0.7 \cdot \exp(-0.5 \cdot (s_1 + s_2)) + 0.2.\end{aligned}$$

For testing purposes, we opted for the Spherical covariance function defined as

$$C(h) = \sigma^2 \cdot C_0\left(\frac{h}{\lambda}\right)$$

where σ^2 and λ are variance and range parameters, respectively, and

$$C_0(h) = \begin{cases} 1 - \frac{3}{2}h + \frac{1}{2}h^3, & \text{if } h < 1,\\ 0, & \text{if } h \geq 1.\end{cases}$$

We multiplied spatially the varying variance with the above covariance function to generate the nonstationary testing datasets. We created 10,000 stationary and 10,000 nonstationary datasets for training and 1000 datasets each for testing. We generated all the datasets on a 10×10 grid.

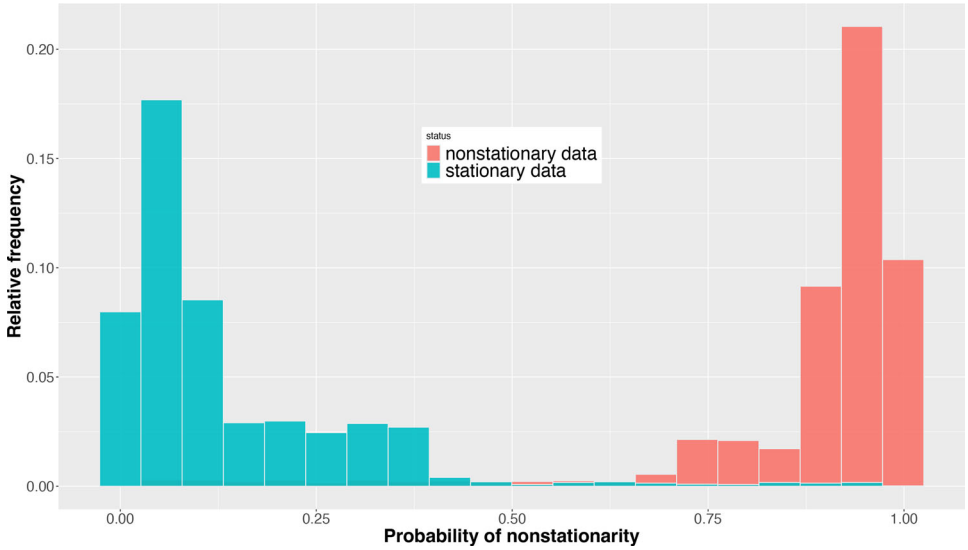

Figure 3. Histograms of the nonstationarity index for stationary testing data and nonstationary testing data in the first simulation setting.

4.2. ConvNet Model Accuracy Assessment

4.2.1. Assessment on First Simulation Setting

We used 12,800 data samples from the stationary and nonstationary regime to train our ConvNet model. Subsequently, we evaluated the model's accuracy by testing it on the remaining datasets, consisting of 3200 stationary and 3200 nonstationary samples generated during the data generation phase.

Figure 3 presents two histograms. The left histogram represents the accuracy of our proposed ConvNet model when applied to stationary data. It demonstrates that the model can successfully detect stationarity (low probability of nonstationarity) in 97.3% of the provided datasets. The right histogram showcases the model's exceptional ability to accurately identify nonstationary spatial data (high probability of nonstationarity), achieving a high percentage of 98.4%. Overall, our trained model achieved an impressive accuracy of 97.9% in correctly classifying the stationary and nonstationary datasets.

4.2.2. Assessment on Second Simulation Setting

We used 10,000 data samples from both the stationary and nonstationary datasets to train our ConvNet model. Subsequently, we evaluated the accuracy of the model by testing it on the generated testing datasets, comprising 100 stationary and 100 nonstationary samples produced during the data generation phase.

Figure 4 displays two histograms. The left histogram illustrates the accuracy of our proposed ConvNet model when applied to stationary data. It reveals that the model can effectively detect stationarity (low probability of nonstationarity) in 91% of the provided datasets. The right histogram demonstrates the model's capability to accurately identify nonstationary spatial data (high probability of nonstationarity), achieving 70% accuracy. The overall accuracy of the model on validation was 80%. It is worth noting that we trained the model on datasets generated on only a 10×10 grid. Despite the low image resolution, the model could still identify the stationarity/nonstationarity and provide high accuracy in validation sets. Our model achieved fair precision when applied to test datasets that had not been seen before. The test accuracy on the nonstationary dataset is low; however, it is important to consider that we only had a model with spatially varying variance, and distinguishing between stationarity and nonstationarity in such scenarios can be challenging.

4.2.3. Comparison With Tests of Stationarity

In recent years, statistical tests assessing the stationarity of the random field, such as Guan, Sherman, and Calvin (2004), Jun and Genton (2012), or Bandyopadhyay and Rao (2017), have gained popularity. However, these statistical tests usually have limitations in preserving the statistical properties. For instance, Guan, Sherman, and Calvin's (2004) test is an isotropy test and cannot distinguish stationary but not isotropic data. Jun and Genton's (2012) test needs to divide the data into two parts and compare the empirical covariance function between the two parts. The division is determined by the user, and this test does not consider more than two divisions. Bandyopadhyay and Rao's (2017) method needs the data to be observed on a rectangular grid, and they did not specify a method for expanding the data in cases where this condition is not met. In a recent work Tzeng, Chen, and Huang (2024) implemented the fused lasso technique to elucidate nonstationary patterns at various resolutions for a single realization of data at irregularly spaced locations. However, in their study, they do not go beyond 2000 spatial locations.

In this comparison, we computed the test of stationarity as proposed by Bandyopadhyay and Rao (2017) on the Spherical covariance-based datasets as discussed in Section 4.1.2. In this test, we choose the tuning parameters $\mathcal{S}, \mathcal{S}'$ and the weight function $g(\cdot)$ of the test statistic using the same method as in Section 6 of Bandyopadhyay and Rao (2017). Reproducible codes are provided at *https://github.com/kaust-es/Spatial_classifier_DL.git*. To check the accuracy of the tests, we computed the p-values for both stationary and nonstationary data and chose the Type-I error to be 0.05. For the stationary scenario, we obtained 98% accuracy, whereas our ConvNet model had 91%. However, on

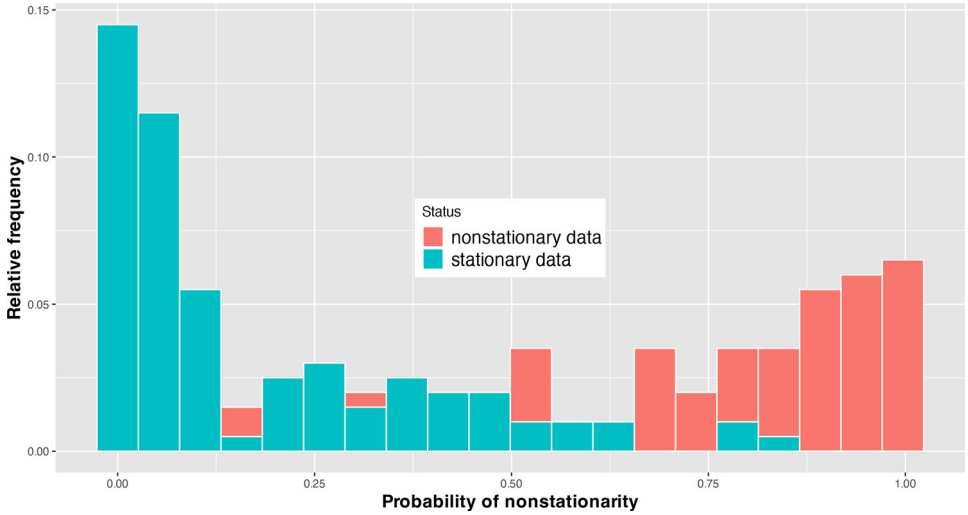

Figure 4. Histograms of the nonstationarity index for stationary testing data and nonstationary testing data in second simulation setting.

Table 1. Average Mean Square Error (MSE) and Standard Error (SE) for parameter estimates in the different simulation settings.

	Method	Number of subregions	MSE_σ ($\times 10^{-4}$)	SE_σ ($\times 10^{-3}$)	MSE_λ ($\times 10^{-5}$)	SE_λ ($\times 10^{-4}$)	MSE_ν ($\times 10^{-5}$)	SE_ν ($\times 10^{-3}$)
Setting 1	ConvNet	3	4.305	2.461	3.322	1.488	1.084	3.511
	ConvNet	2	85.12	3.97	4.171	2.034	866.2	3.721
	User-defined	3	36.86	4.432	3.869	1.733	281.1	4.242
	User-defined	2	120.5	7.788	15.72	11.89	1430.7	17.39
Setting 2	ConvNet	2	234.3	371.3	4.689	1.049	18.51	1.637
	user-defined	2	279.8	439.2	7.078	2.549	612.3	2.835

the nonstationary datasets, the accuracy was just 0.05%, far smaller than our model's accuracy (70%). One of the reasons for this behavior is that Bandyopadhyay and Rao's (2017) test statistic involves a variance estimator, so this test is more suitable for identifying nonstationarity in spatial processes whose variance is constant, which is not the case in our scenario. Hence, this comparison shows that our deep-learning-based method is more flexible than these statistical methods.

4.3. Parameter Estimation Quality

Once the ConvNet model is trained, it can detect low-probability nonstationary regions. This information can then be used to split spatial regions to obtain the representative anchor locations from detected subregions. We rely on the `ExaGeoStat` software to estimate the parameters at all anchor locations. To evaluate the performance of the parameters estimation based on the given subregions, we conducted simulations in three different settings, focusing on estimating three parameters: $\sigma(\boldsymbol{s})$, $\lambda(\boldsymbol{s})$, and $\nu(\boldsymbol{s})$. For each simulation scenario, we present the average values of the parameter estimates across replicated simulations.

In the first simulation scenario (Setting 1), we generated data with four subregions using the nonstationary Matérn covariance function $C^{NS}(\cdot,\cdot;\boldsymbol{\theta})$. The datasets were simulated on an irregular grid of 10,000 points in the spatial domain $[0,1]^2$. The spatially varying parameter regions were defined using kernel smoothing, as explained in (3). We selected four anchor locations: $\mathbf{S}_1 = (0.25, 0.25)^\top$, $\mathbf{S}_2 = (0.25, 0.75)^\top$, $\mathbf{S}_3 = (0.75, 0.25)^\top$, and $\mathbf{S}_4 = (0.75, 0.75)^\top$, with corresponding parameters: $\sigma(\mathbf{S}_1) = 1.2$, $\lambda(\mathbf{S}_1) = 0.05$, $\nu(\mathbf{S}_1) = 0.9$, $\sigma(\mathbf{S}_2) = 0.8$, $\lambda(\mathbf{S}_2) = 0.02$, $\nu(\mathbf{S}_2) = 0.4$, $\sigma(\mathbf{S}_3) = 0.8$, $\lambda(\mathbf{S}_3) = 0.02$, $\nu(\mathbf{S}_3) = 0.4$, $\sigma(\mathbf{S}_4) = 0.8$, $\lambda(\mathbf{S}_4) = 0.02$, and $\nu(\mathbf{S}_4) = 0.4$. The constant ϕ was set to be $\pi/2$ for all simulation scenarios. We estimated the parameters using two and three subregions in this scenario, employing user-defined and ConvNet-based partitioning. The user-defined approach divided the regions equally into two or three subregions by splitting along the x-axis, with the anchor points chosen as the centers of these regions. On the other hand, the anchor points for ConvNet-based subregions were determined using Algorithm 1, as discussed in Section 3.4. Both approaches have been implemented in the `ExaGeoStat` package and can be used to obtain parameter estimates based on exact likelihood computations. Compared to the true values, the average mean squared error (MSE) of spatially varying parameter estimates is presented in Table 1. Different methods and predefined numbers of subregions were employed for this analysis. The table demonstrates that ConvNet subregions yield smaller MSE and standard error (SE) values than user-defined approaches. Furthermore, using a number of subregions close to the true number of subregions reduces estimation errors. Additionally, Figure 5 in Setting 1 presents the heatmap of the parameter estimates averaged over all simulations. The results indicate that the three ConvNet-based subregions outperform the user-defined approach and accurately capture the true patterns of the parameters. Notably, even when the number of subregions was misspecified, the estimated parameters could still capture the true parameter patterns effectively.

In the second simulation scenario (Setting 2), we follow the same formulation as the first one. However, this time we

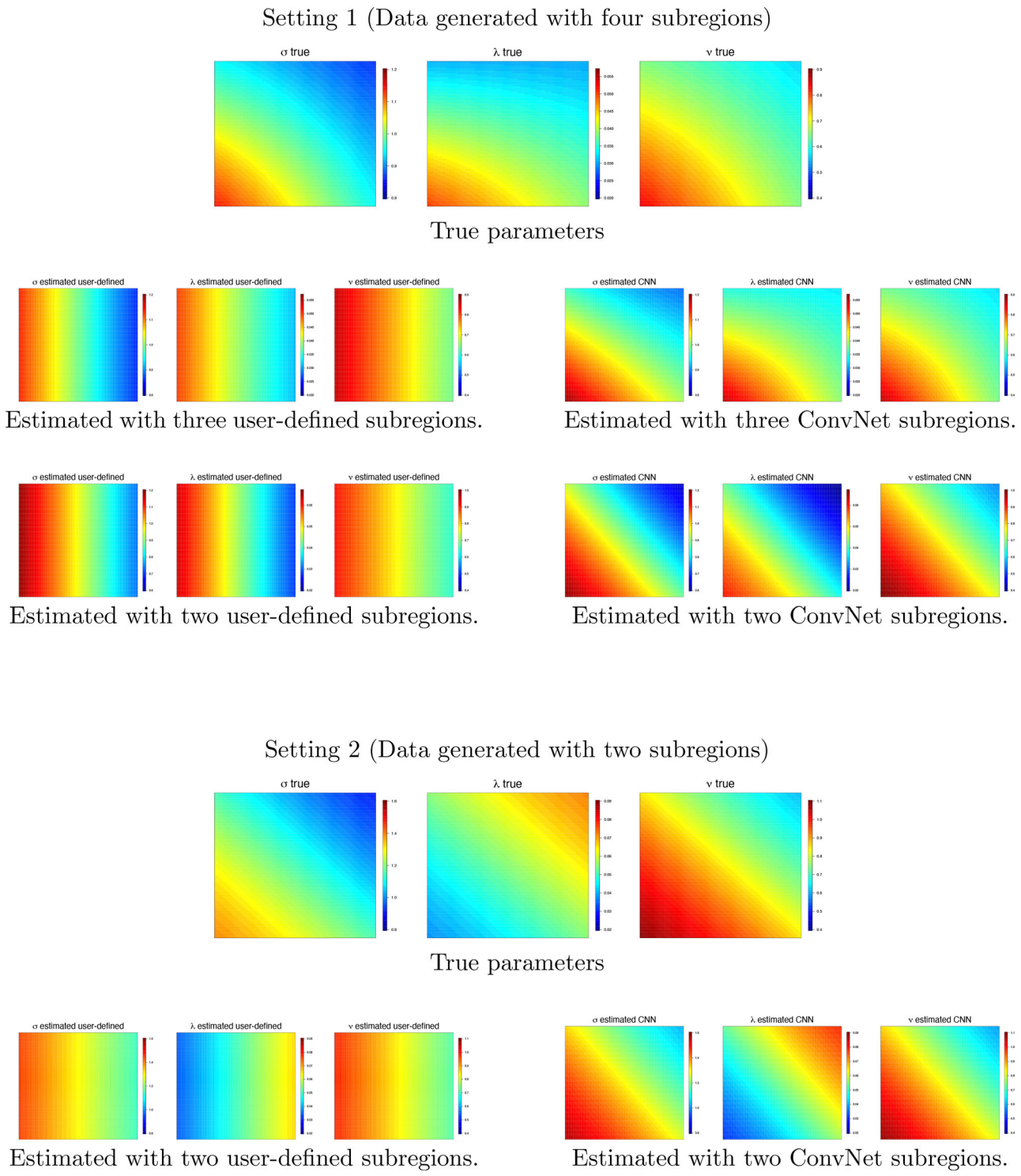

Figure 5. Heatmaps for true parameters and the average of the estimated parameters for different simulation scenarios.

generate data with two subregions and estimate the parameters using two subregions using ConvNet subregions and user-defined subregions methods. This scenario focuses on comparing the efficiency of Algorithm 1 with the user-defined splitting algorithm. We select two anchor points for this setting: $\boldsymbol{S}_1 = (0.25, 0.25)^\top$ and $\boldsymbol{S}_2 = (0.75, 0.75)^\top$. The corresponding parameter values are $\sigma(\boldsymbol{S}_1) = 1.6$, $\lambda(\boldsymbol{S}_1) = 0.09$, $\nu(\boldsymbol{S}_1) = 1.1$, $\sigma(\boldsymbol{S}_2) = 0.8$, $\lambda(\boldsymbol{S}_2) = 0.02$, and $\nu(\boldsymbol{S}_2) = 0.4$. Using the user-defined subregion approach, we split the entire region into two halves along the x-axis. Table 1 shows that the parameter estimates are superior when using the ConvNet subregions compared to the user-defined splitting approach. This can also be observed in Figure 5, where it is clear that the user-defined subregions led to the incorrect selection of anchor points, causing the model to fail in capturing the true spatial pattern of the parameters. However, the ConvNet-based algorithm adaptively selected the subregions, resulting in better parameter estimation.

Table 2. Comparison of the average of the estimated parameters with the true parameter settings for simulation Setting 3.

Parameters	σ	λ	ν
True parameters	2	0.15	0.8
Three ConvNet subregions	2.235	0.137	0.813
	2.077	0.159	0.798
	2.891	0.164	0.814
Four ConvNet subregions	1.983	0.172	0.821
	1.837	0.155	0.796
	2.391	0.178	0.761
	1.939	0.128	0.793

In the third simulation setting (Setting 3), we generated constant parameter values to assess whether our implementation could capture stationarity in parameters. We estimated the constant parameters using three subregions and four subregions, each with 100 replicates. Table 2 presents the parameter estimation results, averaged over all subregions and replicates. It is evident that, on average, the model can estimate the parameter values very close to the true values. Additionally, the parameter estimates from each subregion are close to each other, indicating that the model can efficiently estimate stationary parameter spaces.

5. Soil Moisture Data Application

This section focuses on applying our method to analyze soil moisture content data across the Mississippi Basin region in the United States. The data used in this study were sourced from the HydroBlocks numerical land surface model (Chaney, Metcalfe, and Wood 2016). The study area encompasses approximately 2,400,000 square kilometers, spanning longitudes from −107.7166 to −92.47494 and latitudes from 32.37106 to 43.43772. Due to the varied topography within the region, the soil moisture content exhibits nonstationarity. Huang and Sun (2018) analyzed this dataset with a 1-kilometer resolution. They first removed a spatial mean function and then fitted a stationary Gaussian process to the transformed residuals. In this study, we conduct the same exploratory data analysis but only consider a coarser resolution of 10 kilometers for those residuals and fit the proposed nonstationary Gaussian process. In addition, our analysis focuses on a subset of 200,000 locations selected explicitly for training purposes. As a preprocessing step, we apply a zero-mean Gaussian process model with a Matérn covariance function.

We employ a mean-zero Gaussian likelihood with a nonstationary Matérn covariance, incorporating three spatially varying parameters $\sigma(\mathbf{s}), \lambda(\mathbf{s}), \nu(\mathbf{s})$, and set $\phi = \pi/2$. We assess the parameter estimations using three and four ConvNet subregions, respectively. For these extensive computations, we leverage the Shaheen-II supercomputing facility at KAUST. Shaheen-II is a Cray XC40 system with 6174 dual-socket compute nodes based on 16-core Intel Haswell processors running at 2.3 GHz, where each node has 128 GB of DDR4 memory. The Shaheen-II system has 197,568 processor cores and 790 TB of aggregate memory. We used 256 nodes to do our experiments on the real dataset. The MLE optimization process lasted approximately 17.53 hr and 21.62 hr for the three-cluster and four-cluster configurations, respectively.

The parameter estimates obtained using the nonstationary Matérn covariance model applied to the soil moisture data are depicted in Figure 6. The divisions in each subregion are visibly linear rather than nonlinear. This linearity arises because the subregions are formed using a minimum distance criterion from the node location. Since the dataset is structured on a regular grid, a linear border based on distance will always be generated. We also computed the Akaike Information Criterion (AIC) to evaluate the performance of the models. AIC for three subregions was −274,390.7, and for four subregions, it was −273,408.3. It is evident from the findings that the three subregion modeling framework produces the most accurate parameter estimations for the soil moisture data.

6. Discussion

In this study, we present an approach for modeling nonstationary phenomena in spatial data at large-scale by using dynamic partitioning with the help of ConvNet. Our framework offers a standalone solution that assigns a probability to each spatial region, enabling the differentiation between stationary and nonstationary processes. To evaluate the effectiveness of our framework, we conducted an individual assessment using both stationary and nonstationary datasets generated from the `ExaGeoStat` software. Furthermore, we integrated our ConvNet framework into `ExaGeoStat`, enabling dynamic partitioning of a given spatial region and selecting partitions with low stationary probabilities. This integration improves modeling capabilities under the assumption of nonstationarity.

Herein, we rely on kernel convolution to generate parameter spaces when estimating nonstationary parameters. This approach distinguishes itself from traditional techniques such as weighted local stationary and local stationary methods. Instead of independently calculating the likelihood for each subregion, we adopted a ConvNet data-driven framework where we computed the likelihood for the entire dataset while selecting subregions. As a result, we can visualize the spatially varying parameter as a smooth function in space by estimating the parameters at representative node points. The proposed ConvNet framework has been explicitly trained with Matérn stationary and nonstationary fields for Gaussian likelihood. However, it is important to note that this training only covers a limited range of potential spatial processes. Therefore, future research has room for expansion by training the framework with various simulation types to distinguish between stationary and nonstationary processes even more effectively.

As part of future research, it is possible to replace the ConvNet model discussed in Section 3.1 with graph neural networks (GNNs) (Scarselli et al. 2008; Zhou et al. 2020). Real-world objects often have inherent connections to other entities, and these relationships can be naturally represented as a graph. GNNs, developed over a decade, are neural networks specifically designed to operate on graph data (Scarselli et al. 2008). In recent years, GNNs have been successfully applied in various domains such as antibacterial discovery (Stokes et al. 2020), physics simulations (Sanchez-Gonzalez et al. 2020), fake news detection (Monti et al. 2019), traffic prediction (Nabi et al. 2023), and recommendation systems (Eksombatchai et al. 2018). GNNs have also been used to model spatial processes, including deep Gaussian Markov random fields (Oskarsson, Sidén, and

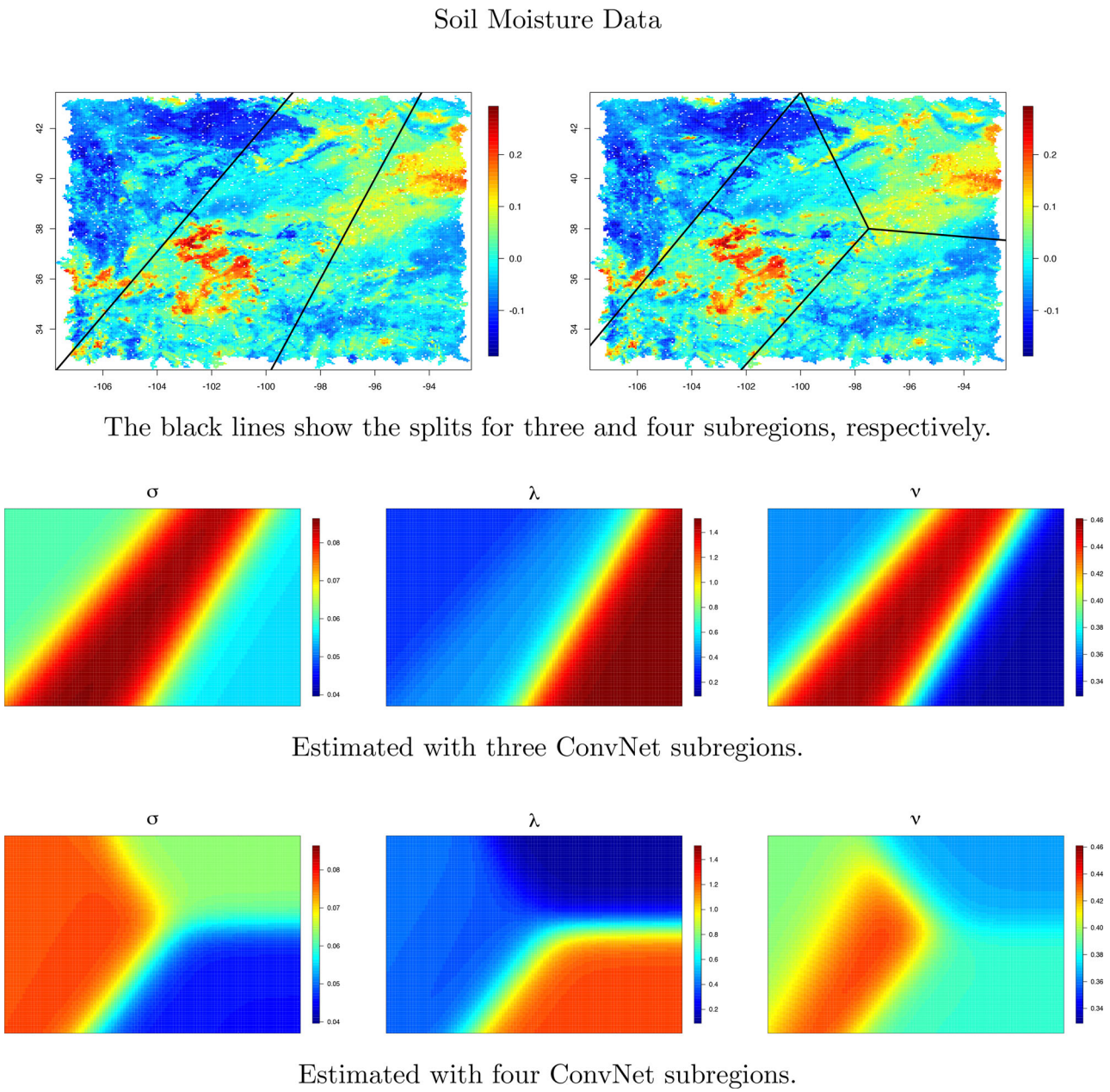

Figure 6. Heatmap of the soil moisture data with its spatially varying parameter estimates over three and four subregion partitions, respectively.

Lindsten 2022). In the context of this study, ConvNets can be easily extended to accommodate graphical structures (Quek et al. 2011), enabling the classification of random processes using GNNs. This integration would contribute to developing a robust framework capable of capturing any spatial pattern, regardless of its regularity or irregularity. Additionally, this approach would reduce the necessity for extensive data pre-processing, as discussed in Section 3.2.

Acknowledgments

We thank the KAUST Supercomputing Laboratory (KSL) for providing computational resources on the Shaheen-II Cray XC40 Supercomputer.

Disclosure Statement

No potential conflict of interest was reported by the author(s).

Funding

The King Abdullah University of Science and Technology (KAUST) in Thuwal, Saudi Arabia, funded the research presented in this article.

ORCID

Pratik Nag http://orcid.org/0000-0001-5065-5273
Ying Sun http://orcid.org/0000-0001-6703-4270

References

Abdulah, S., Li, Y., Cao, J., Ltaief, H., Keyes, D. E., Genton, M. G., and Sun, Y. (2023), "Large-Scale Environmental Data Science with ExaGeoStatR," *Environmetrics*, 34, e2770. [685]

Abdulah, S., Ltaief, H., Sun, Y., Genton, M. G., and Keyes, D. E. (2018a), "ExaGeoStat: A High Performance Unified Software for Geostatistics on Manycore Systems," *IEEE Transactions on Parallel and Distributed Systems*, 29, 2771–2784. [684,685,687]

Abdulah, S., Ltaief, H., Sun, Y., Genton, M. G., and Keyes, D. E. (2018b), "Parallel Approximation of the Maximum Likelihood Estimation for the Prediction of Large-Scale Geostatistics Simulations," in 2018 *IEEE International Conference on Cluster Computing (CLUSTER)*, pp. 98–108, IEEE. [685]

Ajmal, H., Rehman, S., Farooq, U., Ain, Q. U., Riaz, F., and Hassan, A. (2018), "Convolutional Neural Network based Image Segmentation: A Review," *Pattern Recognition and Tracking XXIX*, 10649, 191–203. [685]

Anderes, E. B., and Stein, M. L. (2008), "Estimating Deformations of Isotropic Gaussian Random Fields on the Plane," *The Annals of Statistics*, 36, 719–741. [683,685]

——— (2011), "Local Likelihood Estimation for Nonstationary Random Fields," *Journal of Multivariate Analysis*, 102, 506–520. [684]

Bandyopadhyay, S., and Rao, S. S. (2017), "A Test for Stationarity for Irregularly Spaced Spatial Data," *Journal of the Royal Statistical Society*, Series B, 79, 95–123. [690,691]

Chaney, N. W., Metcalfe, P., and Wood, E. F. (2016), "HydroBlocks: A Field-Scale Resolving Land Surface Model for Application Over Continental Extents," *Hydrological Processes*, 30, 3543–3559. [693]

Cressie, Noel (2015), *Statistics for Spatial Data*, Hoboken, NJ: Wiley. [684]

Eksombatchai, C., Jindal, P., Liu, J. Z., Liu, Y., Sharma, R., Sugnet, C., Ulrich, M., and Leskovec, J. (2018), "Pixie: A System for Recommending 3+ billion Items to 200+ million Users in Real-Time," in *Proceedings of the 2018 World Wide Web Conference*, pp. 1775–1784. [693]

Fouedjio, F., Desassis, N., and Rivoirard, J. (2016), "A Generalized Convolution Model and Estimation for Non-stationary Random Functions," *Spatial Statistics*, 16, 35–52. [684]

Gerber, F., and Nychka, D. (2021), "Fast Covariance Parameter Estimation of Spatial Gaussian Process Models Using Neural Networks," *Stat*, 10, e382. [684]

Guan, Y., Sherman, M., and Calvin, J. A. (2004), "A Nonparametric Test for Spatial Isotropy Using Subsampling," *Journal of the American Statistical Association*, 99, 810–821. [690]

Higdon, D. (1998), "A Process-Convolution Approach to Modelling Temperatures in the North Atlantic Ocean," *Environmental and Ecological Statistics*, 5, 173–190. [684,685]

Huang, H., and Sun, Y. (2018), "Hierarchical Low Rank Approximation of Likelihoods for Large Spatial Datasets," *Journal of Computational and Graphical Statistics*, 27, 110–118. [693]

Jun, M., and Genton, M. G. (2012), "A Test for Stationarity of Spatio-Temporal Random Fields on Planar and Spherical Domains," *Statistica Sinica*, 22, 1737–1764. [690]

Jun, M., and Stein, M. L. (2008), "Nonstationary Covariance Models for Global Data," *The Annals of Applied Statistics*, 2, 1271–1289. [683]

LeCun, Y., Bengio, Y., and Hinton, G. (2015), "Deep Learning," *Nature*, 521, 436–444. [684]

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989), "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Computation*, 1, 541–551. [684]

Lenzi, A., Bessac, J., Rudi, J., and Stein, M. L. (2023), "Neural Networks for Parameter Estimation in Intractable Models," *Computational Statistics & Data Analysis*, 185, 107762. [684]

Lenzi, A., and Rue, H. (2023), "Towards Black-Box Parameter Estimation," arXiv preprint arXiv:2303.15041. [684]

Li, Y., and Sun, Y. (2019), "Efficient Estimation of Nonstationary Spatial Covariance Functions with Application to High-Resolution Climate Model Emulation," *Statistica Sinica*, 29, 1209–1231. [684,685]

Lindgren, F., Rue, H., and Lindström, J. (2011), "An Explicit Link between Gaussian Fields and Gaussian Markov Random Fields: The Stochastic Partial Differential Equation Approach," *Journal of the Royal Statistical Society*, Series B, 73, 423–498. [683]

MacKay, D. J. C. (1998), "Introduction to Gaussian Processes," *NATO ASI Series F Computer and Systems Sciences*, 168, 133–166. [685]

Monti, F., Frasca, F., Eynard, D., Mannion, D., and Bronstein, M. M. (2019), "Fake News Detection on Social Media Using Geometric Deep Learning," arXiv preprint arXiv:1902.06673. [693]

Nabi, S. T., Islam, M. R., Alam, M. G. R., Hassan, M. M., AlQahtani, S. A., Aloi, G., and Fortino, G. (2023), "Deep Learning based Fusion Model for Multivariate LTE Traffic Forecasting and Optimized Radio Parameter Estimation," *IEEE Access*, 11, 14533–14549. [693]

Oskarsson, J., Sidén, P., and Lindsten, F. (2022), "Scalable Deep Gaussian Markov Random Fields for General Graphs," in *International Conference on Machine Learning*, pp. 17117–17137, PMLR. [694]

Paciorek, C., and Schervish, M. (2003), "Nonstationary Covariance Functions for Gaussian Process Regression," in *Advances in Neural Information Processing Systems* (Vol. 16). [684,685]

Paciorek, C. J., and Schervish, M. J. (2006), "Spatial Modelling Using a New Class of Nonstationary Covariance Functions," *Environmetrics*, 17, 483–506. [683,684]

Patterson, H. D. (1975), "Maximum Likelihood Estimation of Components of Variance," in *Proceeding Eight International Biometric Conference, 1975*. Biometric Soc. [685]

Qadir, G. A., Sun, Y., and Kurtek, S. (2021), "Estimation of Spatial Deformation for Nonstationary Processes via Variogram Alignment," *Technometrics*, 63, 548–561. [683]

Quek, A., Wang, Z., Zhang, J., and Feng, D. (2011), "Structural Image Classification with Graph Neural Networks," in *2011 International Conference on Digital Image Computing: Techniques and Applications*, pp. 416–421, IEEE. [694]

Rawat, W., and Wang, Z. (2017), "Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review," *Neural Computation*, 29, 2352–2449. [685]

Risser, M. D. (2016), "Review: Nonstationary Spatial Modeling, with Emphasis on Process Convolution and Covariate-Driven Approaches," arXiv:1610.02447. [683]

Risser, M. D., and Calder, C. A. (2017), "Local Likelihood Estimation for Covariance Functions with Spatially-Varying Parameters: The convoSPAT Package for R," *Journal of Statistical Software*, 81, 1–32. [684]

Sampson, P. D., and Guttorp, P. (1992), "Nonparametric Estimation of Nonstationary Spatial Covariance Structure," *Journal of the American Statistical Association*, 87, 108–119. [683]

Sanchez-Gonzalez, A., Godwin, J., Pfaff, T., Ying, R., Leskovec, J., and Battaglia, P. (2020), "Learning to Simulate Complex Physics with Graph Networks," in *International Conference on Machine Learning*, pp. 8459–8468, PMLR. [693]

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. (2008), "The Graph Neural Network Model," *IEEE Transactions on Neural Networks*, 20, 61–80. [693]

Stein, M. L. (2005), "Nonstationary Spatial Covariance Functions," unpublished technical report. [684,685]

Stokes, J. M., Yang, K., Swanson, K., Jin, W., Cubillos-Ruiz, A., Donghia, N. M., MacNair, C. R., French, S., Carfrae, L. A., Bloom-Ackermann, Z., et al. (2020), "A Deep Learning Approach to Antibiotic Discovery," *Cell*, 180, 688–702. [693]

Sun, Y., Li, B., and Genton, M. G. (2012), "Geostatistics for Large Datasets," in *Advances and Challenges in Space-time Modelling of Natural Events*, eds. E. Porcu, J.-M. Montero, and M. Schlather, pp. 55–77, Berlin, Heidelberg: Springer. [685]

Tzeng, S., Chen, B.-Y., and Huang, H.-C. (2024), "Assessing Spatial Stationarity and Segmenting Spatial Processes into Stationary Components," *Journal of Agricultural, Biological and Environmental Statistics*, 29, 301–319. [690]

Vecchia, A. V. (1988), "Estimation and Model Identification for Continuous Spatial Processes," *Journal of the Royal Statistical Society*, Series B, 50, 297–312. [685]

Whittle, P. (1954), "On Stationary Processes in the Plane," *Biometrika*, 41, 434–449. [685]

Wilson, A., and Adams, R. (2013), "Gaussian Process Kernels for Pattern Discovery and Extrapolation," in *International Conference on Machine Learning*, pp. 1067–1075, PMLR. [683]

Zhang, K., Zhang, Z., Li, Z., and Qiao, Y. (2016), "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks," *IEEE Signal Processing Letters*, 23, 1499–1503. [684]

Zhiqiang, W., and Jun, L. (2017), "A Review of Object Detection based on Convolutional Neural Network," in *2017 36th Chinese Control Conference (CCC)*, pp. 11104–11109, IEEE. [685]

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., and Sun, M. (2020), "Graph Neural Networks: A Review of Methods and Applications," *AI Open*, 1, 57–81. [693]