JYX JYVÄSKYLÄN YLIOPISTO UNIVERSITY OF JYVÄSKYLÄ

This is a self-archived version of an original article. This version may differ from the original in pagination and typographic details.

Author(s): Sipilä, Mika; Nordhausen, Klaus; Taskinen, Sara

Title: Nonlinear blind source separation exploiting spatial nonstationarity

Year: 2024

Version: Published version

Copyright: © 2024 the Authors

Rights: CC BY 4.0

Rights url: https://creativecommons.org/licenses/by/4.0/

Please cite the original version:

Sipilä, M., Nordhausen, K., & Taskinen, S. (2024). Nonlinear blind source separation exploiting spatial nonstationarity. Information Sciences, 665, Article 120365. https://doi.org/10.1016/j.ins.2024.120365

Information Sciences 665 (2024) 120365

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Nonlinear blind source separation exploiting spatial nonstationarity

Mika Sipilä *, Klaus Nordhausen, Sara Taskinen

Department of Mathematics and Statistics, University of Jyvaskyla, Finland

ARTICLE INFO

Keywords:
Independent component analysis
Multivariate spatial data
Shapley values
Variational autoencoder

ABSTRACT

In spatial blind source separation the observed multivariate random fields are assumed to be mixtures of latent spatially dependent random fields. The objective is to recover latent random fields by estimating the unmixing transformation. Currently, the algorithms for spatial blind source separation can only estimate linear unmixing transformations. Nonlinear blind source separation methods for spatial data are scarce. In this paper, we extend an identifiable variational autoencoder that can estimate nonlinear unmixing transformations to spatially dependent data, and demonstrate its performance for both stationary and nonstationary spatial data using simulations. In addition, we introduce scaled mean absolute Shapley additive explanations for interpreting the latent components through nonlinear mixing transformation. The spatial identifiable variational autoencoder is applied to a geochemical dataset to find the latent random fields, which are then interpreted by using the scaled mean absolute Shapley additive explanations. Finally, we illustrate how the proposed method can be used as a pre-processing method when making multivariate predictions.

1. Introduction

Nowadays spatially indexed data are encountered in many fields of science. For example, in geochemistry, samples are collected at different locations and analyzed for their chemical compositions to detect patterns, for identifying areas of pollution, or for mining purposes. From now on, we denote the location as $\boldsymbol{s} \in \mathcal{S} \subset \mathbb{R}^q$, and let $\boldsymbol{x}(\boldsymbol{s}) = (x_1(\boldsymbol{s}), \dots, x_d(\boldsymbol{s}))^\top$ be the observable d-variate vector. Here the set of possible locations $\mathcal{S}$ is the domain and the data are usually called spatial data. In most applications $q = 2$ and for simplicity, without loss of generality, we also assume the same from now on. Spatial data are usually characterized by its mean function $\boldsymbol{m}(\boldsymbol{s})$ and a (spatial) covariance function $\boldsymbol{C}(\boldsymbol{x}(\boldsymbol{s}), \boldsymbol{x}(\boldsymbol{s}')) = E\left((\boldsymbol{x}(\boldsymbol{s}) - \boldsymbol{m}(\boldsymbol{s}))(\boldsymbol{x}(\boldsymbol{s}') - \boldsymbol{m}(\boldsymbol{s}'))^\top\right)$, where $\boldsymbol{s}, \boldsymbol{s}' \in \mathcal{S}$. The main idea in modeling is that observations closer together are more similar than observations further apart. However, as the covariance function is usually quite complex, modeling spatial data is often quite difficult and is, for example, much more challenging than modeling time series data, which have a natural direction (past to present) that is missing in spatial data. Thus, $\boldsymbol{x}(\boldsymbol{s})$ is quite often assumed as stationary to simplify spatial modeling, meaning that $\boldsymbol{m}(\boldsymbol{s}) = \boldsymbol{m}$ for all $\boldsymbol{s} \in \mathcal{S}$ and $\boldsymbol{C}(\boldsymbol{x}(\boldsymbol{s}), \boldsymbol{x}(\boldsymbol{s}')) = \boldsymbol{C}(||\boldsymbol{s} - \boldsymbol{s}'||)$ for all $\boldsymbol{s}, \boldsymbol{s}' \in \mathcal{S}$, i.e. the mean is the same at all locations and the spatial dependence depends only on the distance between observations. For an overview discussing the complexity of modeling $\boldsymbol{C}$, see [1].

Another suggestion for simplifying spatial data modeling is considering a latent component modeling as a pre-processing step. The data are then represented using d independent, latent components in $\boldsymbol{z}(\boldsymbol{s}) = (z_1(\boldsymbol{s}), \dots, z_d(\boldsymbol{s}))^\top$. Thus, after pre-processing, univariate

* Corresponding author.
E-mail address: mika.e.sipila@jyu.fi (M. Sipilä).

https://doi.org/10.1016/j.ins.2024.120365
Received 13 November 2023; Received in revised form 9 January 2024; Accepted 21 February 2024

Available online 28 February 2024
0020-0255/© 2024 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

models can be fitted to components and the components can be interpreted individually and predicted separately [2]. While the latent components can simply be seen as useful modeling tools, they often turn out to have physical meanings for which however subject knowledge is usually needed. A popular approach to latent component modeling is linear blind source separation (BSS) [3], which aims to find independent latents components using the observed data by assuming that the observations are generated from the components through some linear mixing system. Recently, [4–6] suggested a spatial BSS (SBSS) approach for stationary multivariate spatial data, where the independent latent components are obtained by jointly diagonalizing two or more moment-based matrices. SBSS was also extended to nonstationary multivariate spatial data by [7], then yielding nonstationary spatial source separation (SNSS). In the SNSS model, $\boldsymbol{x}(\boldsymbol{s})$ is assumed to be stationary with respect to $\boldsymbol{m}$ but not with respect to $\boldsymbol{C}$. A drawback of SBSS and SNSS is that both assume a linear mixing procedure, which is interpretable and mathematically tractable but might be too simplistic in reality.

The motivation for the use of nonlinear BSS approach is the same as for linear BSS, that is, one wants to find a useful representation of high-dimensional data to be used in further analyses [8]. As the mixing procedure is assumed to be nonlinear, the components are unidentifiable without additional assumptions on the mixing function or on the distribution of the latent components [9]. Recently, unsupervised deep learning methods, such as variational autoencoders (VAEs) [10], and their extensions, such as [11,12], were applied to find unmixing function and latent components based on the observed data. As the regular VAE solutions in general suffer from unidentifiability issues, [13] proposed in a time series context the identifiable VAE (iVAE) method, which uses some additionally observed data to make the VAE model identifiable. In [14] a structured nonlinear BSS framework, which is suitable also for two dimensional graphs, was proposed. However, to our knowledge, models for spatial nonlinear BSS have not been developed or studied yet.

In this paper, we extend the iVAE method for spatially dependent data and study its performance using vast simulation studies and a real geochemical data example. Nonlinear BSS framework in general lacks tools for latent component interpretation, which is an essential task when analyzing the discovered latent representation. Therefore, a procedure based on Shapley additive explanations [15] for interpreting the latent components through nonlinear mixing transformation is introduced and illustrated. The paper is organized as follows. We review the basics of linear and nonlinear BSS in Section 2. Sections 3 and 4 introduce new iVAE method and new tools for interpreting the latent components, respectively. In Sections 5 and 6, simulation studies and a real data example are used to illustrate the performance properties of iVAE. The paper is concluded with a discussion in Sections 7 and 8.

2. Background

Before showing how iVAE can be extended to spatial data, we give some background on linear and nonlinear BSS as well as Shapley values, which are later used to interpret the latent components.

2.1. Linear and nonlinear BSS

Assume for now that $\boldsymbol{x} = (x_1, \dots, x_d)^\top$ is a d-variate observable random vector. In linear blind source separation (BSS), one assumes that $\boldsymbol{x}$ is a linear mixture of unknown d-variate latent components $\boldsymbol{z} = (z_1, \dots, z_d)^\top$, that is,

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{z}, \tag{1}$$

where $\boldsymbol{A}$ is an invertible $d \times d$ mixing matrix. The goal in BSS is to recover both $\boldsymbol{A}$ and $\boldsymbol{z}$ based on $\boldsymbol{x}$ only, but this cannot be done without making some additional assumptions on $\boldsymbol{z}$. Different BSS methods vary in the assumption made on latent components. For example, in independent component analysis (ICA) we assume iid data with independent latent components, and in second-order source separation (SOS) we assume that the components are weakly stationary time series. For general overview of BSS methods, see [3]. In [5], spatial BSS methods were introduced.

As the assumption on linear mixing is often too restrictive, nonlinear BSS methods have been introduced. For a recent review of methods, see [8]. Notice that although the methodology is often referred to as nonlinear ICA, in what follows, we use the more general term, nonlinear BSS, as it allows the components to be of any data type. In nonlinear BSS, one assumes that $\boldsymbol{x}$ is generated by applying an unknown invertible mixing transformation $\boldsymbol{f} : \mathbb{R}^d \to \mathbb{R}^d$ on the latent components $\boldsymbol{z}$ as

$$\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{z}). \tag{2}$$

The objective is then to identify the transformation $\boldsymbol{q} : \mathbb{R}^d \to \mathbb{R}^d$ which returns $\boldsymbol{z}$ as

$$\boldsymbol{z} = \boldsymbol{q}(\boldsymbol{x}) \tag{3}$$

based on the observations of the vector $\boldsymbol{x}$ only. Generally, without additional assumptions on the mixing function or on the distribution of the latent components, nonlinear BSS is unidentifiable as there is an infinite amount of nonlinear transformations to generate mutually independent components from the observations [9]. Thus, additional constraints on the distribution of $\boldsymbol{z}$ must be introduced to obtain identifiability without restricting the mixing transformation $\boldsymbol{f}$ severely.

In many recent studies [16–18,13,19], the main assumption leading to identifiability is that each latent component z_i, $i = 1, \dots, d$, is statistically dependent on an m-dimensional auxiliary variable $\boldsymbol{u}$, and that the conditional distribution is a member of the family of exponential distributions with parameters $\lambda_{i,k}$, $k = 1, \dots, r$. Assuming the independence of the d conditional latent distributions yields then the joint distribution $p(\boldsymbol{z}|\boldsymbol{u}) = \prod_{i=1}^{d} p(z_i|\boldsymbol{u})$. Following [13], the dependence of the parameters $\lambda_{i,k}$ on $\boldsymbol{u}$ is expressed using

a function $\lambda(\boldsymbol{u}) = (\lambda_1(\boldsymbol{u}), \dots, \lambda_d(\boldsymbol{u}))$, where $\lambda_i(\boldsymbol{u}) = (\lambda_{i,1}(\boldsymbol{u}), \dots, \lambda_{i,r}(\boldsymbol{u}))^\top$ contains the parameters depending on $\boldsymbol{u}$ for component i. This function must be learned, for example, using some neural network. The joint conditional distribution can then be written as

$$p_{T,\lambda}(\boldsymbol{z}|\boldsymbol{u}) = \prod_{i=1}^{d} \frac{Q_i(z_i)}{Z_i(\boldsymbol{u})} \exp\left[\sum_{k=1}^{r} T_{i,j}(z_i)\lambda_{i,k}(\boldsymbol{u})\right], \tag{4}$$

where $Q_i(z_i)$ is a base measure, $Z_i(\boldsymbol{u})$ is a normalizing constant, and $\boldsymbol{T}_i(z_i) = (T_{i,1}(z_i), \dots, T_{i,r}(z_i))^\top$ contains sufficient statistics. The dimension r of each sufficient statistic $\boldsymbol{T}_i(z_i)$ and $\lambda_i(\boldsymbol{u})$ is assumed to be fixed. By assuming the source distribution (4) and the additionally observed auxiliary variable $\boldsymbol{u}$, the aim becomes identifying the latent sources using the unmixing transformation $\boldsymbol{q}(\boldsymbol{x}, \boldsymbol{u})$.

In a time series context, there are several examples of auxiliary variables found in the literature. In the case of stationary time series, one can use as $\boldsymbol{u}$, for example, the previous observations [17]. For nonstationary time series data, the time segment of the observation can be used [16,13,19]. We propose in Section 3 a nonlinear BSS method for spatial data. The method extends the method proposed in [13] that uses VAEs with auxiliary variables to learn the full identifiable generative model, and is thus referred as identifiable VAE (iVAE).

2.2. SHAP values and mean absolute SHAP values

The main advantage of using nonlinear BSS methods is that they can discover some meaningful underlying latent representation through a nonlinear mixing environment by recovering the true latent components $\boldsymbol{z}$. However, the interpretation of the latent representation can be difficult without knowing how the observed variables interact with the latent components. In case of linear BSS, it is easy to inspect the contributions through mixing/unmixing matrices, but when the mixing function is nonlinear, we do not have such methods available. Later in this paper, we suggest novel metrics for evaluating contributions of input variables for nonlinear models with multiple outputs, such as iVAE. We explain the procedure of calculating these values in detail in Section 4, but let us first review here the Shapley values [20] and Shapley additive explanations (SHAP) [15] that our metrics base upon.

The SHAP framework is based on the Shapley values, which are originally used in cooperative game theory to fairly distribute the payout of a game between a group of players. Let $\mathcal{K}$ be a set of K players and γ be a function giving the payout of the game for any set of players K. The Shapley values calculated based on the function γ produce the attributions a_i for each player $x_i \in \mathcal{K}$ so that

$$\gamma(A) = a_0 + \sum_{i=1}^{K} a_i x_i', \tag{5}$$

where $A \subseteq \mathcal{K}$, a_0 is a baseline output without any players and $x_i' = \mathbb{1}(x_i \in A)$ is a binary variable which gets a value of 1 if x_i belongs in set A and value of 0 otherwise. The function $\mathbb{1}$ is an indicator function. The Shapley value for player x_i is then given by

$$a_i = \sum_{A \subseteq \mathcal{K} \setminus x_i} \frac{|A|!(|\mathcal{K}| - |A| - 1)!}{K!} (\gamma(A \cup x_i) - \gamma(A)), \tag{6}$$

where $|\cdot|$ denotes the size of a set. The game can be looked as any mathematical function. Then, the function inputs correspond the players, and the output of the function corresponds the payload. If the game γ is a statistical model, the players are the observations of K variables $\mathcal{K} = \{x_1, \dots, x_K\}$ and the payout is the predicted value of a variable y produced by γ with an input $A \subseteq \mathcal{K}$, then the attribution of an observation x_i of ith variable to the outcome prediction is given by a_i.

Typically, statistical models have fixed input size, which is why the formulation in (6) is not directly applicable. The SHAP framework suggests a procedure where $\gamma(A)$ is estimated with $E(\tilde{\gamma}(\boldsymbol{x})|A)$, where $\boldsymbol{x} = (x_1, \dots, x_K)^\top$ and $\tilde{\gamma}$ has a fixed input size of K. The Shapley values calculated based on these conditional expected values $E(\tilde{\gamma}(\boldsymbol{x})|A)$ are called SHAP values, mathematically given by

$$\tilde{a}_i = \sum_{A \subseteq \mathcal{K} \setminus x_i} \frac{|A|!(|\mathcal{K}| - |A| - 1)!}{K!} (E(\tilde{\gamma}(\boldsymbol{x})|A \cup x_i) - E(\tilde{\gamma}(\boldsymbol{x})|A)). \tag{7}$$

The conditional expected value $E(\tilde{\gamma}(\boldsymbol{x})|A)$ is calculated in practice by averaging over the outputs of $\tilde{\gamma}$ of a representative background data while keeping the observed values in set A fixed. While it is possible to calculate the exact SHAP values using this procedure, the computational burden becomes heavy when the number of variables is high. To reduce the computational burden, the kernel SHAP method [15] can be used. The kernel SHAP approximates the SHAP values $\tilde{a}_i$ based on local interpretable model-agnostic explanations (LIME) [21] by minimizing the equation

$$\zeta = \sum_{A \subseteq \mathcal{K}} \left(E(\tilde{\gamma}(\boldsymbol{x})|A) - \tilde{a}_0 + \sum_{i=1}^{K} \tilde{a}_i x_i' \right)^2 \pi(A), \tag{8}$$

where

$$\pi(A) = \frac{K - 1}{\binom{K}{|A|} |A|(K - |A|)}, \tag{9}$$

with respect to the values $\tilde{a}_i$. By using the specific kernel $\pi(A)$ called Shapley kernel, the algorithm in fact recovers the Shapley values. For more details, see [15].

The SHAP values are calculated individually for each (multivariate) observation/prediction pair making them a useful tool to interpret how each component of the observation affects the model prediction. In many cases, we are interested in the population level effects, that is, how the observed variables affect the model prediction on population level. Methods, such as mean absolute SHAP (MASHAP) [22] and Shapley global additive importance (SAGE) [23], aim to obtain the population level importance for the input variables. In this paper, we focus on MASHAP values, which can be easily extended to models with multiple outputs as will be shown in Section 4. Let now n be the number of observations and d the number of observable variables, and let $\tilde{a}_{i,j}$ be a SHAP value for ith variable of jth observation. The MASHAP value for the ith variable is then calculated as

$$v_i = \frac{1}{n}\sum_{j=1}^{n} |\tilde{a}_{i,j}|. \tag{10}$$

The MASHAP values v_i can be interpreted as contribution or importance of ith input variable to the whole output population.

3. Nonstationary spatial source separation using VAE

In this paper, we use iVAE [13] to solve the nonlinear SNSS problem by assuming nonstationarity in the variance of the latent fields. We use spatial segmentation as an auxiliary variable $\boldsymbol{u}$ and assume that the sources are distributed as in (4). With spatial segmentation we mean that the domain $\mathcal{S}$ is divided into m segments $\mathcal{S}_i \subset \mathcal{S}$, so that $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$ for all $i \neq j$ and $\cup_{i=1}^{m} \mathcal{S}_i = \mathcal{S}$, $i, j = 1, \dots, m$. Then, the auxiliary variable for the observation $\boldsymbol{x}(\boldsymbol{s})$ is $\boldsymbol{u}(\boldsymbol{s}) = (\mathbb{1}(\boldsymbol{s} \in \mathcal{S}_1), \dots, \mathbb{1}(\boldsymbol{s} \in \mathcal{S}_m)))^\top$, i.e. a m-dimensional standard basis vector giving the segment corresponding to the spatial location of the observation. From now on, we use the notation $\boldsymbol{x} := \boldsymbol{x}(\boldsymbol{s})$, $\boldsymbol{z} := \boldsymbol{z}(\boldsymbol{s})$, and $\boldsymbol{u} := \boldsymbol{u}(\boldsymbol{s})$ for the sake of convenience. iVAE assumes the following generating model:

$$p_{\boldsymbol{f},\boldsymbol{T},\lambda}(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{u}) = p_{\boldsymbol{f}}(\boldsymbol{x}|\boldsymbol{z}) p_{\boldsymbol{T},\lambda}(\boldsymbol{z}|\boldsymbol{u}), \tag{11}$$

where $p_{\boldsymbol{f}}(\boldsymbol{x}|\boldsymbol{z})$ is defined as

$$p_{\boldsymbol{f}}(\boldsymbol{x}|\boldsymbol{z}) = p_{\epsilon_{\boldsymbol{f}}}(\boldsymbol{x} - \boldsymbol{f}(\boldsymbol{z})), \tag{12}$$

which means that $\boldsymbol{x}$ can be composed of an independent noise vector $\epsilon_{\boldsymbol{f}}$ and a mixing transformation $\boldsymbol{f}(\boldsymbol{z})$ as $\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{z}) + \epsilon_{\boldsymbol{f}}$. The distribution $p_{\epsilon_{\boldsymbol{f}}}$ is assumed to be a zero mean Gaussian distribution with infinitesimal variance to match the form of the nonnoisy nonlinear ICA model in (2). In the limit of infinite data, iVAE recovers the true latent components $\boldsymbol{z}$ up to permutation and signed scaling under some mild conditions on the mixing function $\boldsymbol{f}$, the sufficient statistics $\boldsymbol{T}$ and the auxiliary variable $\boldsymbol{u}$ [13]. Here, we focus on Gaussian distributed latent variables, in which case the variances of the latent components are required to vary enough based on the auxiliary variable $\boldsymbol{u}$, and the mixing function $\boldsymbol{f}$ is required to have continuous partial derivatives.

iVAE consists mainly of the encoder $\boldsymbol{g}$, the decoder $\boldsymbol{h}$ and the auxiliary function $\boldsymbol{w}$. The encoder $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}) = (\boldsymbol{g}_{\boldsymbol{\mu}}(\boldsymbol{x}, \boldsymbol{u})^\top, \boldsymbol{g}_{\boldsymbol{\sigma}}(\boldsymbol{x}, \boldsymbol{u})^\top)^\top$ maps the observed data $(\boldsymbol{x}, \boldsymbol{u})$ to mean and variance vectors $\boldsymbol{\mu}_{\boldsymbol{z}|\boldsymbol{x},\boldsymbol{u}}$ and $\boldsymbol{\sigma}_{\boldsymbol{z}|\boldsymbol{x},\boldsymbol{u}}$. The mean and the variance are used to sample a new latent representation $\boldsymbol{z}'$ by using a reparametrization trick [10], which is generating $\boldsymbol{z}'$ as $\boldsymbol{z}' = \boldsymbol{\mu}_{\boldsymbol{z}|\boldsymbol{x},\boldsymbol{u}} + \boldsymbol{\sigma}_{\boldsymbol{z}|\boldsymbol{x},\boldsymbol{u}}^\top \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim N(\boldsymbol{0}, \boldsymbol{I})$ and $\boldsymbol{I}$ is an identity matrix. The estimate of the unmixing transformation $\boldsymbol{q}$ is obtained as $\boldsymbol{g}_{\boldsymbol{\mu}}$. This means that we obtain the latent components $\boldsymbol{z}$ as $\boldsymbol{\mu}_{\boldsymbol{z}|\boldsymbol{x},\boldsymbol{u}}$, which is a mean of the variational distribution given by the encoder. Meanwhile, the decoder $\boldsymbol{h}(\boldsymbol{z})$ estimates the true mixing function $\boldsymbol{f}(\boldsymbol{z})$ and maps the latent representation $\boldsymbol{z}'$ back to observable data $\boldsymbol{x}'$. The auxiliary function $\boldsymbol{w}(\boldsymbol{u})$ estimates the parameters $\lambda(\boldsymbol{u})$ of (4) in the segment given by $\boldsymbol{u}$. The functions $\boldsymbol{g}, \boldsymbol{h}$ and $\boldsymbol{w}$ are deep neural networks, which are flexible function estimators and we denote their parameters as $\boldsymbol{\theta}_{\boldsymbol{g}}, \boldsymbol{\theta}_{\boldsymbol{h}}$, and $\boldsymbol{\theta}_{\boldsymbol{w}}$, respectively, which must be estimated. Coming back to the generating model as in (11), we now have that

$$p_{\boldsymbol{\theta}_{\boldsymbol{h}},\boldsymbol{\theta}_{\boldsymbol{w}}}(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{u}) = p_{\boldsymbol{\theta}_{\boldsymbol{h}}}(\boldsymbol{x}|\boldsymbol{z}) p_{\boldsymbol{\theta}_{\boldsymbol{w}}}(\boldsymbol{z}|\boldsymbol{u}), \tag{13}$$

where $p_{\boldsymbol{\theta}_{\boldsymbol{h}}}(\boldsymbol{x}|\boldsymbol{z})$ is defined as in (12). iVAE learns simultaneously the full latent generative model and the variational approximation $q_{\boldsymbol{\theta}_{\boldsymbol{g}}}(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{u})$ of its true distribution $p_{\boldsymbol{q}}(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{u})$. The parameters $\boldsymbol{\theta} = (\boldsymbol{\theta}_{\boldsymbol{g}}^\top, \boldsymbol{\theta}_{\boldsymbol{h}}^\top, \boldsymbol{\theta}_{\boldsymbol{w}}^\top)^\top$ are estimated by maximizing the lower bound of the data log-likelihood defined by

$$\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{x}, \boldsymbol{u}) \geq E_{q_{\boldsymbol{\theta}_{\boldsymbol{g}}}(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{u})}\left(\log p_{\boldsymbol{\theta}_{\boldsymbol{h}}}(\boldsymbol{x}|\boldsymbol{z}) + \log p_{\boldsymbol{\theta}_{\boldsymbol{w}}}(\boldsymbol{z}|\boldsymbol{u}) - \log q_{\boldsymbol{\theta}_{\boldsymbol{g}}}(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{u})\right). \tag{14}$$

The variational approximation $q_{\boldsymbol{\theta}_{\boldsymbol{g}}}(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{u})$ is obtained using the reparametrization trick. The schematic presentation of iVAE is illustrated in Fig. 1. In our simulation studies in Section 5, we assume the source density distribution $p_{\boldsymbol{\theta}_{\boldsymbol{w}}}(\boldsymbol{z}|\boldsymbol{u})$ to be Gaussian, where the mean and variance vectors, $\boldsymbol{\mu}_{\boldsymbol{z}|\boldsymbol{u}}$ and $\boldsymbol{\sigma}_{\boldsymbol{z}|\boldsymbol{u}}$, are given by the auxiliary function $\boldsymbol{w}(\boldsymbol{u})$. Hence, in practice we have $p_{\boldsymbol{\theta}_{\boldsymbol{w}}}(\boldsymbol{z}|\boldsymbol{u}) = N(\boldsymbol{z}|\boldsymbol{\mu}_{\boldsymbol{z}|\boldsymbol{u}}, \mathrm{diag}(\boldsymbol{\sigma}_{\boldsymbol{z}|\boldsymbol{u}}))$, $q_{\boldsymbol{\theta}_{\boldsymbol{g}}}(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{u}) = N(\boldsymbol{z}|\boldsymbol{\mu}_{\boldsymbol{z}|\boldsymbol{x},\boldsymbol{u}}, \mathrm{diag}(\boldsymbol{\sigma}_{\boldsymbol{z}|\boldsymbol{x},\boldsymbol{u}}))$, and $p_{\boldsymbol{\theta}_{\boldsymbol{h}}}(\boldsymbol{x}|\boldsymbol{z}) = N(\boldsymbol{x}|\boldsymbol{x}', \beta\boldsymbol{I})$, where $\beta > 0$ is a constant close to zero as $p_{\boldsymbol{\theta}_{\boldsymbol{h}}}$ should estimate the true distribution $p_{\boldsymbol{f}}$ with infinitesimal variance. We use $\beta = 0.01$.

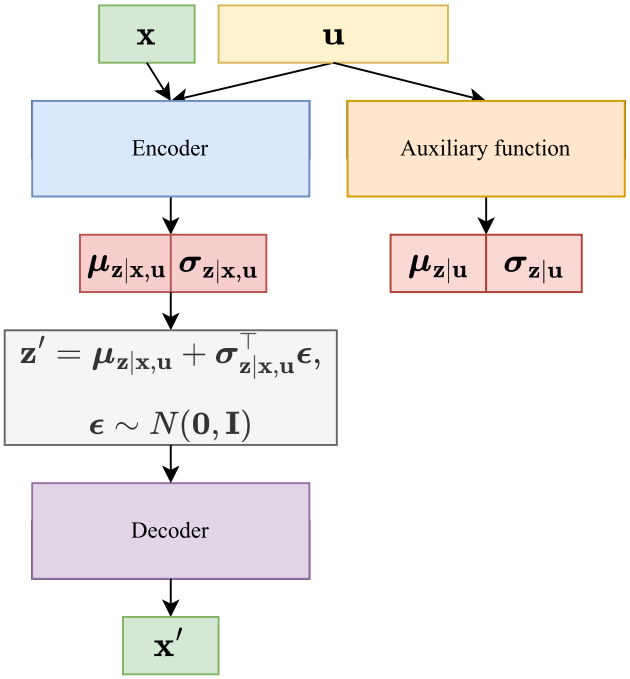

Fig. 1. Schematic presentation of iVAE. iVAE learns simultaneously the encoder, the decoder, and the auxiliary function. The latent components $\boldsymbol{z}$ are obtained as $\boldsymbol{\mu}_{\boldsymbol{z}|\boldsymbol{x},\boldsymbol{u}}$, which is a mean of the variational distribution given by the encoder.

4. Interpreting the latent representation

In this section, we introduce the scaled MASHAP values and explain how they can be used to evaluate variable importance for model with multiple outputs.

MASHAP and SAGE, as defined in Section 2.2, estimate successfully the population level importance of input variables for a single output variable. However, for a model with multiple outputs it would be beneficial to be able to compare the population level importance values across the different outputs, as well as to obtain an importance value for each input variable with respect to all output variables. Here, we suggest a heuristic procedure to obtain scaled MASHAP values, which are comparable importance values between different output variables, and to calculate average scaled MASHAP values, which are interpreted as a population level importance values for the models with multiple output variables. The procedure to obtain scaled MASHAP and average scaled MASHAP values is summarized in Algorithm 1, and discussed in detail next.

Algorithm 1: An algorithm for calculating the scaled MASHAP values and average scaled MASHAP values.

Input: $n \times K$ data matrix $\boldsymbol{X}$, function $\tilde{\boldsymbol{\gamma}}^* : \mathbb{R}^K \to \mathbb{R}^H$;
Output: Scaled MASHAP values $\boldsymbol{V}$; Average scaled MASHAP values $\boldsymbol{v}^*$;
Initialize: $H \times K$ matrix $\boldsymbol{V}$; $1 \times K$ vector $\boldsymbol{v}^*$;
for $i = 1$ **to** H **do**
  Set $\tilde{\gamma} = \tilde{\gamma}^*_k$;
  for $k = 1$ **to** K **do**
    Calculate SHAP values $\tilde{a}_{k,1}, \dots \tilde{a}_{k,n}$ as in (7);
    Calculate MASHAP value v_k as in (10);
    Set $\boldsymbol{V}[i,k] \leftarrow v_k$;
  end
  Set $\boldsymbol{V}[i,] \leftarrow \frac{\boldsymbol{V}[i,]}{\sum_{k=1}^{K} \boldsymbol{V}[i,k]}$;
end
for $k = 1$ **to** K **do**
  Set $\boldsymbol{v}^*[k] \leftarrow \frac{\sum_{i=1}^{H}(\boldsymbol{V}[i,k])}{H}$
end

Let $\boldsymbol{\gamma}^*$ be a model or a function with H-dimensional output. As SHAP and MASHAP assume a function with a single output, we calculate the MASHAP values independently for each output $i = 1, \dots, H$ of the function $\boldsymbol{\gamma}^*$ by setting $\tilde{\gamma} = \gamma^*_i$. Then, we obtain the contributions of each input variable to each output variable. However, by using this procedure, the MASHAP values are not comparable across different input variables. This is because the MASHAP values give only the relative importance of each input

variable to a single output variable. As the distributions and scales of the output variables might differ, the sum of the MASHAP values of the input variables is not constant. Hence, the values are not relative to each other across the output variables, and comparisons of the MASHAP values might lead to false interpretations.

To make the MASHAP values comparable, we suggest scaling the values to unit sum for each individual output variable. The scaling ensures that all output variables are weighted identically making the scaled MASHAP values comparable also across the outputs. In addition, we propose average scaled MASHAP values to identify the importance of the input variables for a model with multiple output variables. Let $\boldsymbol{V}$ be a matrix whose rows $\boldsymbol{v}^k$ contain the scaled MASHAP values for the kth input variable. By taking means column-wise, we get a K-dimensional vector $\boldsymbol{v}^*$ giving the relative importance of each input variable.

In context of nonlinear BSS, the scaled MASHAP values can be used to evaluate the contribution of each observed variable to a latent component, and the contribution of each latent component to an observable variable. By calculating the average scaled MASHAP values for the BSS model's mixing estimate, we obtain importance values for each latent component. When the average scaled MASHAP value is high, the latent component explains high proportion of the original variables, and when it is low, the latent component does not have much impact on most of the variables.

The interpretations based on scaled MASHAP and average scaled MASHAP values are demonstrated in more detail in Section 6.

5. Simulations

While we have the theoretical results of identifiability when the variances of latent components are varying enough based on the auxiliary variable $\boldsymbol{u}$, these results apply only in limit of infinite data. In real life applications, with finite data, there is however no guarantee that the identifiability conditions are fulfilled. In this section, we aim to study the performance of iVAE in various spatial scenarios with finite sample size using the simulation studies. We consider six different simulation settings, where the observed data are generated from nonlinear ICA model (2). The underlying latent components $\boldsymbol{z}$ are generated differently in each setting. Some of the settings exhibit nonstationary variance, meaning that the identifiability conditions are fulfilled, while some settings are stationary, for which the identifiability conditions are not fulfilled. We compare iVAE to SBSS and SNSS, although they are developed for linear mixing and are thus not optimal when the mixing function is nonlinear. In addition, iVAE is compared against a modified version of time contrastive learning (TCL) [16], where the auxiliary variable is a spatial segmentation instead of a time segmentation. TCL exploits nonstationarity in variance when solving the BSS problem and can estimate nonlinear unmixing transformations. The simulations can be reproduced using R 4.3.0 [24] together with R packages SpatialBSS [25], gstat [26] and NonlinearBSS. The NonlinearBSS package, which is available at https://github.com/mikasip/NonlinearBSS, contains an R implementation of spatial iVAE and some methods to generate nonstationary spatial data from the nonlinear ICA model (2). The simulations were executed on the CSC Puhti cluster, a high-performance computing environment.

In all simulation settings, $n = 5000$ locations $\boldsymbol{s}_j$, $j = 1, \ldots, n$, are sampled uniformly in the spatial domain $\mathcal{S} = [0, 100] \times [0, 100]$. For each location $\boldsymbol{s}_j$ a three-variate observation $\boldsymbol{x}_j = \boldsymbol{x}(\boldsymbol{s}_j)$ is generated, meaning that $d = 3$ in all simulations. In Settings 1, 2 and 3, the data consist of ten clusters and are generated using different parameters in each cluster. The clusters are generated by sampling ten random points and dividing the field so that each cluster contains the locations that have the smallest distance to the corresponding point. Notice that we do not use the ground truth segmentation as an auxiliary variable in modeling as was done in previous studies, such as [16–18,13], in a time series context. Instead, we use small enough segments to allow the model to approximate the ground truth clusters without having prior information on them. This is illustrated in Fig. 2, where $n = 5000$ uniformly sampled points in ten clusters are presented. In the figure, the colors represent the ten generated ground truth clusters and the grid on top of the plot illustrates the spatial segmentation used by iVAE and TCL.

Settings 1 and 2. The latent field consists of identically and independently distributed (iid) three-variate Gaussian vectors. Each cluster, $k = 1, \ldots, 10$, has a unique, diagonal covariance matrix $\boldsymbol{\Sigma}_k = \text{diag}(\sigma_{1,k}, \sigma_{2,k}, \sigma_{3,k})$, where $\sigma_{1,k}, \sigma_{2,k}, \sigma_{3,k} \sim \text{Unif}(0.1, 5)$. In Settings 1 and 2 the mean vectors are $\boldsymbol{\mu}_k = (0, 0, 0)^\top$ and $\boldsymbol{\mu}_k = (\mu_{1,k}, \mu_{2,k}, \mu_{3,k})^\top$, respectively, and $\mu_{1,k}, \mu_{2,k}, \mu_{3,k} \sim \text{Unif}(-5, 5)$.

Setting 3. The latent field consists of three-variate Gaussian fields with a Matern correlation structure. Each cluster has its own parameters for the Matern covariance function

$$C_M(h; \nu, \phi) = \frac{1}{2^{\nu-1}\Gamma(\nu)} \left(\frac{h}{\phi}\right)^{\nu} K_\nu \left(\frac{h}{\phi}\right), \tag{15}$$

where $h = ||\boldsymbol{s} - \boldsymbol{s}'||$ is the Euclidean distance between two locations $\boldsymbol{s}$ and $\boldsymbol{s}'$, $\nu > 0$ is a scale parameter, $\phi > 0$ is a range parameter, Γ is the gamma function, and K_ν is the modified Bessel function of the second kind with shape parameter ν. The range and scale parameters are generated independently from $\nu_i \sim \text{Unif}(0.1, 5)$ and $\phi_i \sim \text{Unif}(0.5, 8)$.

Settings 4 and 5. The latent field is a single three-variate Gaussian random field with a Matern correlation structure. The Matern parameters for Setting 4 are $(\nu_1, \phi_1) = (0.5, 15), (\nu_2, \phi_2) = (2, 20)$, and $(\nu_3, \phi_3) = (0.2, 10)$. In Setting 5, the parameters are $(\nu_1, \phi_1) = (1, 5), (\nu_2, \phi_2) = (2, 3)$, and $(\nu_3, \phi_3) = (6, 2)$. Setting 4 shows strong spatial dependence and higher variability in sample variance across the space, whereas Setting 5 shows weak spatial dependence and low variability in sample variance.

Setting 6. The latent field is a single three-variate random field with a nonstationary Matern correlation structure as presented in [27]:

$$\begin{aligned} &C_{M_n}(\boldsymbol{s}, \boldsymbol{s}'; \sigma, \nu, \phi) = \sigma(\boldsymbol{s})\sigma(\boldsymbol{s}')r_1(\boldsymbol{s})r_1(\boldsymbol{s}')r_2(\boldsymbol{s}, \boldsymbol{s}')^{-1} \\ &\left(r_2(\boldsymbol{s}, \boldsymbol{s}')^{-1/2}h\right)^{\nu(\boldsymbol{s},\boldsymbol{s}')} K_{\nu(\boldsymbol{s},\boldsymbol{s}')}(r_2(\boldsymbol{s}, \boldsymbol{s}')^{-1/2}h), \end{aligned} \tag{16}$$

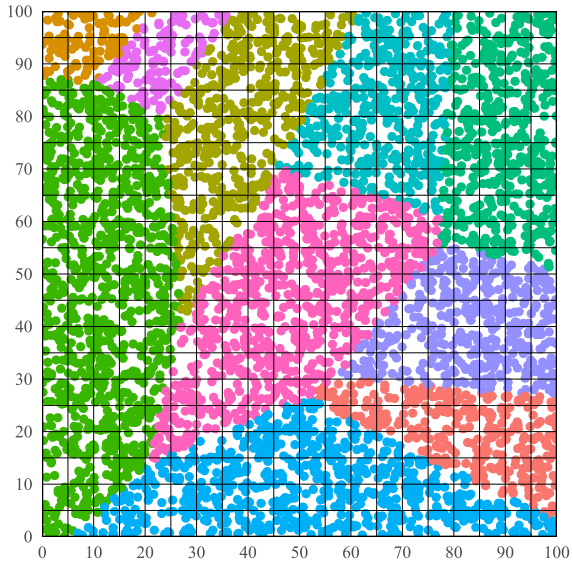

Fig. 2. Five thousand simulated points in a 100×100 field based on ten randomly generated clusters. The colors indicate true cluster membership and the grid on top of the plot illustrates the spatial segmentation used by TCL and iVAE.

where $h = ||\boldsymbol{s} - \boldsymbol{s}'||$ and we denote $\nu(\boldsymbol{s}, \boldsymbol{s}') = (\nu(\boldsymbol{s}) + \nu(\boldsymbol{s}'))/2$,

$$r_1(\boldsymbol{s}) = \left(\frac{\phi^2(\boldsymbol{s})/4\nu(\boldsymbol{s})}{\Gamma(\nu(\boldsymbol{s}))2^{\nu(\boldsymbol{s})-1}} \right)^{1/2} \quad \text{and}$$
$$r_2(\boldsymbol{s}, \boldsymbol{s}') = \left(\frac{\phi^2(\boldsymbol{s})}{8\nu(\boldsymbol{s})} + \frac{\phi^2(\boldsymbol{s}')}{8\nu(\boldsymbol{s}')} \right), \tag{17}$$

where $\sigma : \mathbb{R}^2 \to \mathbb{R}^+$, $\nu : \mathbb{R}^2 \to \mathbb{R}^+$ and $\phi : \mathbb{R}^2 \to \mathbb{R}^+$ are the local variance, shape, and range parameter functions, respectively. We choose these functions as

$$\sigma(\boldsymbol{s}; \boldsymbol{d}_\sigma, \alpha_\sigma) = \log(1.1 + (\boldsymbol{s}^\top \boldsymbol{d}_\sigma)/\alpha_\sigma),$$
$$\nu(\boldsymbol{s}; \boldsymbol{d}_\nu, \alpha_\nu) = (\boldsymbol{s}^\top \boldsymbol{d}_\nu)^{1/5}/\alpha_\nu + 0.1, \tag{18}$$
$$\phi(\boldsymbol{s}; \boldsymbol{d}_\phi, \alpha_\phi, c_\phi) = (\boldsymbol{s}^\top \boldsymbol{d}_\phi)/\alpha_\phi + c_\phi).$$

For the first component z_1, we select the parameters $\boldsymbol{d}_\sigma = (1,1)$, $\alpha_\sigma = 2$, $\boldsymbol{d}_\nu = (0,1)$, $\alpha_\nu = 5$, $\boldsymbol{d}_\phi = (1,1)$, $\alpha_\phi = 5$, and $c_\phi = 10$; for z_2, we select $\boldsymbol{d}_\sigma = (0,1)$, $\alpha_\sigma = 1.5$, $\boldsymbol{d}_\nu = (1,0)$, $\alpha_\nu = 4$, $\boldsymbol{d}_\phi = (1,1)$, $\alpha_\phi = -8$, and $c_\phi = 40$; for z_3, we select $\boldsymbol{d}_\sigma = (1,0)$, $\alpha_\sigma = 2$, $\boldsymbol{d}_\nu = (1,1)$, $\alpha_\nu = 3$, $\boldsymbol{d}_\phi = (0,1)$, $\alpha_\phi = 4$, and $c_\phi = 10$. By using the above correlation structure, the variance, range, and scale parameters of the random field vary in space differently within each component.

Settings 1 and 2 are considered the easiest settings for the iVAE model as variances (and means in Setting 2) of the latent fields explicitly change between the clusters. These settings are spatial variants of the time series settings of some previous simulation studies, such as [13,16], where the latent components have varying means/variances based on some time segments. The settings aim to set a baseline performance under conditions, where the variances are explicitly changing based on our chosen auxiliary variable. Meanwhile, Settings 4 and 5 are used to measure the performance when the latent fields are stationary, and thus not identifiable in theory. Of these two, Setting 4 has higher variability in sample mean and in sample variance through the latent fields making it more optimal for iVAE. Settings 3 and 6 are chosen to illustrate the performance when the latent fields are nonstationary.

Mixing procedure. The observed data $\boldsymbol{x}$ are generated by applying a mixing function $\boldsymbol{f}_L$ to the latent fields $\boldsymbol{z}$ as $\boldsymbol{x} = \boldsymbol{f}_L(\boldsymbol{z})$. The mixing function $\boldsymbol{f}_L$ is generated using multilayer perceptron (MLP) with L layers following previous studies [16,18,13]. In making the mixing function invertible and differentiable, the number of hidden units in each layer equals the number of latent components and the nonlinear activation functions are smooth bijective functions, such as a hyperbolic tangent or exponential linear unit (ELU) [28]. In addition, the mixing matrices of the layers of MLP are normalized to have unit length row and column vectors to guarantee that none of the latent components vanish in the mixing process. Let ω_i be the activation function of ith layer and $\boldsymbol{B}_i$ be the normalized mixing matrix of ith layer. Then, the mixing function $\boldsymbol{f}_L$ is defined as

$$\boldsymbol{f}_L(\boldsymbol{z}) = \begin{cases} \omega_L(\boldsymbol{B}_L \boldsymbol{z}), & L = 1, \\ \omega_L(\boldsymbol{B}_L \boldsymbol{f}_{L-1}(\boldsymbol{z})), & L \in \{2, 3, \dots\}. \end{cases} \tag{19}$$

In these simulations, we use linear activation $\omega_L(x) = x$ for the last layer and ELU activation

$$\omega_i(x) = \begin{cases} x, & x \geq 0, \\ \exp(x) - 1, & x < 0, \end{cases} \tag{20}$$

$i = 1, \ldots, L-1$, for the other layers. This means that the mixing function $\boldsymbol{f}_1$ with one layer corresponds to a linear mixing case. The simulations are repeated 1000 times with 1, 2 and 3 mixing layers.

Model specifications. In iVAE's encoder, decoder, and auxiliary function, we use three hidden layers with 128 units and leaky rectified linear unit (ReLU) activation to ensure that the model is capable of estimating the possible nonlinearities in mixing/unmixing functions and in the function $\lambda(\boldsymbol{u})$, which is modeled by the auxiliary function. Likewise, in TCL, we use three hidden layers with 128 units and leaky ReLU activation. For both models, iVAE and TCL, a 20×20 segmentation is used as an auxiliary variable, meaning that the spatial domain is divided into $m = 400$ equally sized squares. This segmentation is illustrated in Fig. 2. iVAE and TCL are trained for 200 epochs with a batch size of 64. The initial learning rate is 0.01, which is decreased with second-order polynomial decay over 10000 training steps until the learning rate of 0.0001. SBSS and SNSS algorithms use two ring kernels defined by the parameters $\boldsymbol{r}_1 = (0, 2)$ and $\boldsymbol{r}_2 = (2, 4)$. Further, SNSS uses a 10×10 segmentation. The sizes of the rings and the segmentation for SNSS yielded the best average performance in an initial simulation containing 100 trials of each setting with various segmentations and ring sizes.

Performance index. We use the mean correlation coefficient (MCC), which is a widely used performance metric in nonlinear ICA, to measure the performance of different methods, e.g. [16,17,13,18]. MCC is a function of the correlation matrix $\boldsymbol{K} = \mathrm{Cor}(\hat{\boldsymbol{z}}, \boldsymbol{z})$, where $\hat{\boldsymbol{z}}$ includes the estimated latent components, and is calculated as

$$\mathrm{MCC}(\boldsymbol{K}) = \frac{1}{p} \sup_{\boldsymbol{P} \in \mathcal{P}} \mathrm{tr}(\boldsymbol{P}\,\mathrm{abs}(\boldsymbol{K})), \tag{21}$$

where $\mathcal{P}$ is a set of all possible permutation matrices, $\mathrm{tr}(\cdot)$ is the trace of a matrix, and $\mathrm{abs}(\cdot)$ denotes taking the absolute value of a matrix elementwise. The optimal MCC value is one in which case the estimated and true sources are perfectly correlated up to their signs.

Results. The resulting MCC values are presented in Fig. 3. In Settings 1, 2, 3 and 6, where the variances of the latent fields are varying based on the spatial location, and thus fulfilling the identifiability conditions, iVAE shows superior performance as compared with SBSS, SNSS, and TCL. In Setting 1, where the latent fields are zero-mean Gaussian with varying variances between the clusters, SNSS performs almost as well as iVAE, especially when the number of mixing layers is one. However, in Setting 2, where the mean also changes between the clusters, the performance of SNSS drops considerably, whereas the performance of iVAE remains good. In Setting 3, SBSS and SNSS perform similarly, but the methods do not reach the performance of iVAE, especially when the number of mixing layers is high. Meanwhile, Settings 4 and 5 are stationary and thus, in theory, more favorable for SBSS. Surprisingly, in Setting 4, where the Matern covariance function's range parameters ϕ are high and the scale parameters ν are low, iVAE outperforms SBSS. This might be because such Matern parameters lead to high spatial dependence and larger variability in sample variance compared to Setting 5, which means that the mean and the variance are changing in some degree through out the spatial domain allowing the identifiability. However, in Setting 5, where the parameters are selected so that the mean and the variance are more stable throughout the field, SBSS slightly outperforms iVAE. In Setting 6, iVAE is the only method that is reliably capable of separating the sources. In conclusion, as long as the sample variance has enough variability in space, iVAE recovers the latent components well and outperforms the competing methods.

6. Real data example

In this section we demonstrate the spatial iVAE, SBSS and SNSS methods to a dataset derived from GEMAS geochemical mapping project [29], which is available in the R package robCompositions [30]. The application is two-fold. First, the latent fields are estimated with each method and the latent representations are compared using scaled MASHAP values. As we do not have any ground truth latent fields available as in simulations in Section 5, only the interpretability of the latent representations is compared. Second, we study if the prediction power can be increased by predicting the latent fields to new locations instead of predicting the observed variables directly. The predicted latent fields are then back transformed to the original variables by using the estimated mixing transformations provided by iVAE, SBSS and SNSS. TCL is not applied for the data as it does not estimate the mixing transformation $\boldsymbol{f}$. For this reason, the scaled MASHAP values cannot be utilized for evaluating the importance of the latent components. For the same reason, the method is incapable of back transforming the estimated latent components, and thus cannot be applied for prediction purposes. The code for the analysis of the data is available at https://github.com/mikasip/NonlinearSNSS. The analysis was executed on a laptop running Windows 10, equipped with an Intel Core i5 processor (1.7 GHz) and 16 GB RAM.

The dataset consists of 2108 concentration measurements of 18 chemical elements (Al, Ba, Ca, Cr, Fe, K, Mg, Mn, Na, Nb, P, Si, Sr, Ti, V, Y, Zn, Zr) in agricultural soil samples. We dropped one observation measured in Canary islands as it lacks neighboring observations. Therefore $n = 2107$. The sample locations are presented in Fig. 4.

In this example, we are dealing with compositional data, meaning that the variables carry relative information to other variables rather than absolute information. In this dataset, the measurements are chemical elements' relative proportions of the whole soil sample measured as milligrams per kilogram. Due to the sum constraints, compositional data do not follow the Euclidean geometry and it is custom to transform such data prior analysis. [4] argue that in the context of BSS, the isometric log-ratio (ilr) [31] transformation is a natural choice yielding full rank data. While ilr coordinates are not easy to interpret, they have the additional benefit that they have a one-to-one relationship with the centralized log-ratio (clr) transformation. The clr transformation is popular among

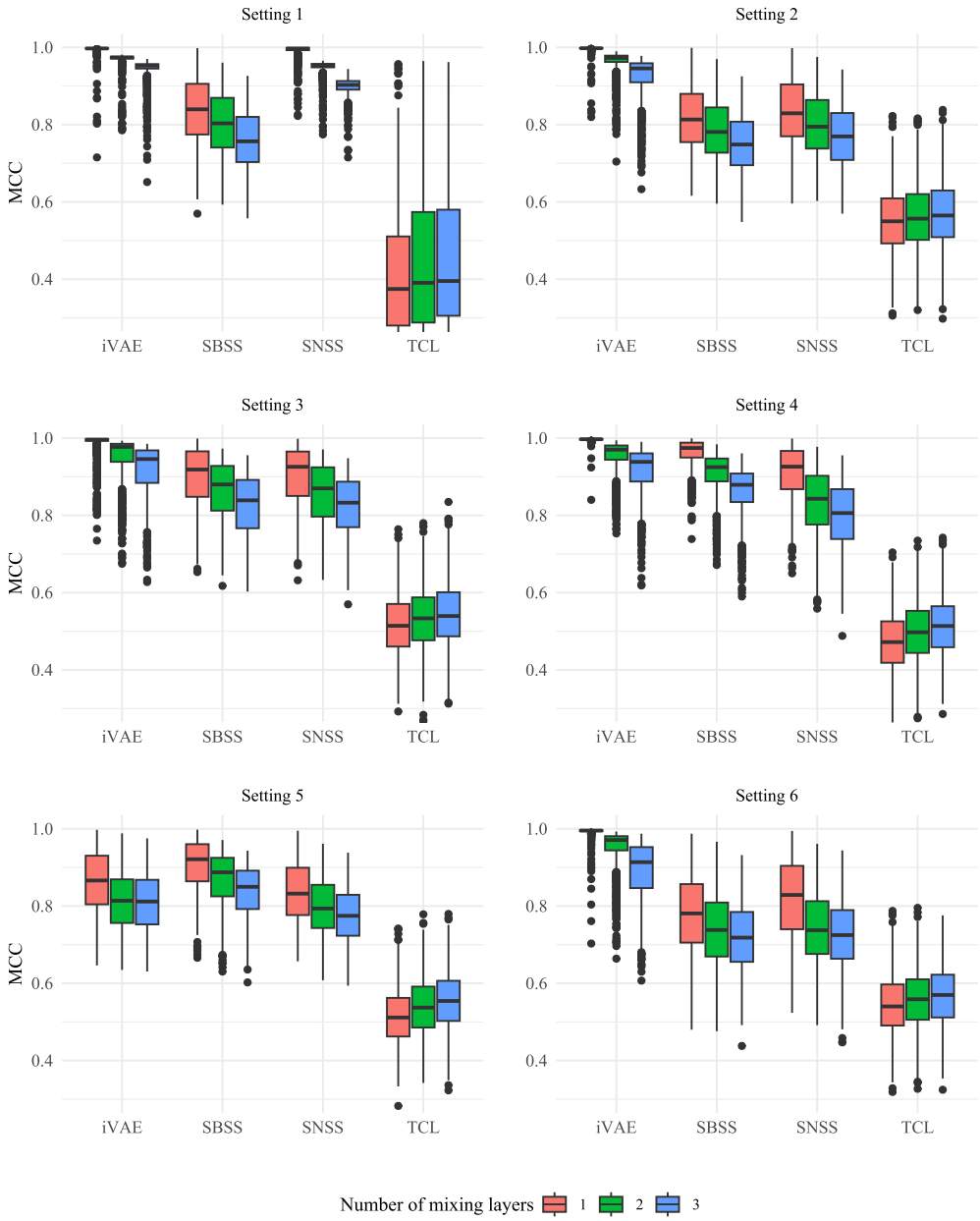

Fig. 3. Boxlots of the mean correlation coefficients (MCC) of 1000 simulation trials of Settings 1-6 for iVAE, SBSS, SNSS and TCL. The color of the boxplots indicates the number of mixing layers where one layer corresponds to linear mixing.

practitioners, because, although the coordinates are singular, they are interpretable. Accordingly, the data are processed using an ilr transformation which reduces the dimension of the dataset by one, and therefore the dimension of the preprocessed data is 17. For interpretation purposes the ilr coordinates are transformed to clr coordinates and their interpretation goes similarly as in the case of log-transformed data, see [31] for further details.

In this application, we use the same iVAE model hyperparameters as in the simulation studies of Section 5. As an auxiliary variable, we use 10 km $\times$ 10 km spatial segmentation, which is illustrated in Fig. 4. Note that all spatial segments with zero observations are dropped. For SBSS and SNSS, we select the parameters provided in [7], where the same dataset was studied. Hence, we use one ring kernel with radius of 167 km for both SBSS and SNSS, and divide the spatial domain to four equally sized square segments for SNSS.

First, the contributions of the latent components to the clr transformed variables are determined for iVAE, SBSS and SNSS by calculating the scaled MASHAP values for each method's mixing function estimate. For mixing function estimate, the input is the estimated latent components and the output is the ilr transformed observations, which are then transformed back in the clr transformed observations. Then, the scaled MASHAP values can be interpreted as importance of each latent feature to an original variable (in log scale). The averages of the scaled MASHAP values can be interpreted as importance of each latent feature to the whole

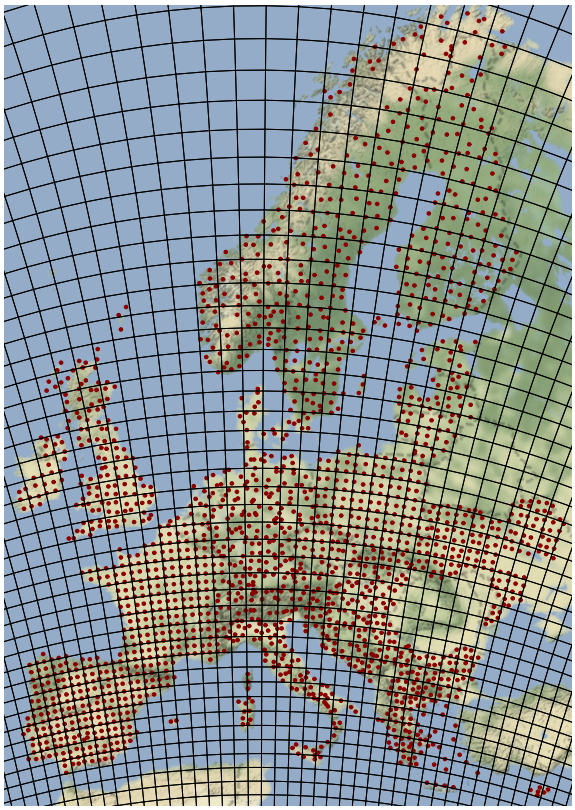

Fig. 4. Sample locations of GEMAS dataset and 100 km grid representing the segmentation used in iVAE model.

Table 1
The scaled MASHAP values for the decoder part of the trained iVAE model.

	IC1	IC2	IC3	IC4	IC5	IC6	IC7	IC8	IC9	IC10	IC11	IC12	IC13	IC14	IC15	IC16	IC17
clr(Al)	0.099	0.119	0.154	0.169	0.093	0.075	0.039	0.041	0.023	0.080	0.049	0.025	0.008	0.007	0.007	0.007	0.007
clr(Ba)	0.223	0.074	0.021	0.163	0.092	0.113	0.035	0.048	0.114	0.053	0.015	0.019	0.009	0.005	0.005	0.005	0.005
clr(Ca)	0.046	0.410	0.016	0.096	0.175	0.051	0.046	0.059	0.030	0.018	0.009	0.023	0.005	0.004	0.004	0.004	0.004
clr(Cr)	0.236	0.044	0.085	0.162	0.038	0.028	0.125	0.026	0.065	0.137	0.013	0.015	0.007	0.005	0.005	0.005	0.005
clr(Fe)	0.404	0.098	0.101	0.060	0.032	0.025	0.034	0.037	0.026	0.045	0.076	0.028	0.009	0.006	0.006	0.006	0.006
clr(K)	0.236	0.148	0.053	0.207	0.052	0.035	0.083	0.022	0.016	0.056	0.049	0.019	0.006	0.004	0.004	0.004	0.004
clr(Mg)	0.197	0.223	0.139	0.086	0.090	0.021	0.033	0.089	0.034	0.045	0.012	0.014	0.004	0.003	0.003	0.003	0.003
clr(Mn)	0.252	0.039	0.174	0.020	0.119	0.178	0.024	0.040	0.111	0.007	0.014	0.010	0.003	0.002	0.002	0.002	0.002
clr(Na)	0.163	0.158	0.020	0.069	0.274	0.155	0.024	0.020	0.021	0.011	0.009	0.054	0.006	0.004	0.004	0.004	0.004
clr(Nb)	0.055	0.338	0.083	0.037	0.031	0.036	0.097	0.095	0.020	0.036	0.097	0.045	0.007	0.006	0.006	0.006	0.006
clr(P)	0.057	0.079	0.251	0.048	0.186	0.161	0.032	0.090	0.016	0.038	0.008	0.020	0.003	0.003	0.003	0.003	0.003
clr(Si)	0.300	0.231	0.055	0.029	0.056	0.024	0.093	0.086	0.029	0.019	0.014	0.036	0.008	0.005	0.005	0.005	0.005
clr(Sr)	0.175	0.366	0.058	0.046	0.033	0.016	0.102	0.019	0.035	0.020	0.018	0.092	0.005	0.004	0.004	0.004	0.004
clr(Ti)	0.099	0.310	0.127	0.087	0.078	0.021	0.074	0.029	0.065	0.022	0.023	0.041	0.006	0.005	0.005	0.005	0.005
clr(V)	0.381	0.081	0.100	0.082	0.026	0.022	0.043	0.017	0.085	0.105	0.014	0.022	0.005	0.005	0.005	0.005	0.005
clr(Y)	0.057	0.290	0.087	0.038	0.087	0.047	0.076	0.073	0.116	0.026	0.058	0.011	0.013	0.005	0.005	0.005	0.005
clr(Zn)	0.237	0.087	0.137	0.163	0.084	0.027	0.047	0.086	0.045	0.056	0.012	0.005	0.004	0.002	0.002	0.002	0.002
clr(Zr)	0.241	0.309	0.026	0.106	0.041	0.045	0.044	0.021	0.021	0.052	0.040	0.025	0.010	0.005	0.005	0.005	0.005
Average	**0.192**	**0.189**	0.094	0.093	0.088	0.060	0.058	0.050	0.049	0.046	0.029	0.028	0.007	0.004	0.004	0.004	0.004

dataset. The scaled MASHAP values are calculated by using 200 observations as background data. To ensure that the background data represents the whole dataset as well as possible, the background locations are selected one by one, so that the new location has the maximum distance to any previously selected location. The obtained background data are presented in Fig. 5.

For interpretation, we order latent components in decreasing order based on the average scaled MASHAP values. By inspecting the resulting scaled MASHAP values of iVAE's mixing function estimate provided in Table 1, we see that the latent components 1 and 2 are the most important ones as they have the highest average scaled MASHAP values of 0.192 and 0.189, whereas the values of other components are smaller than 0.094. The fact, that the average scaled MASHAP values of latent components IC13-IC17 are almost zero, indicates that these components are essentially noise and do not explain any of the chemical elements. Hence, fewer latent

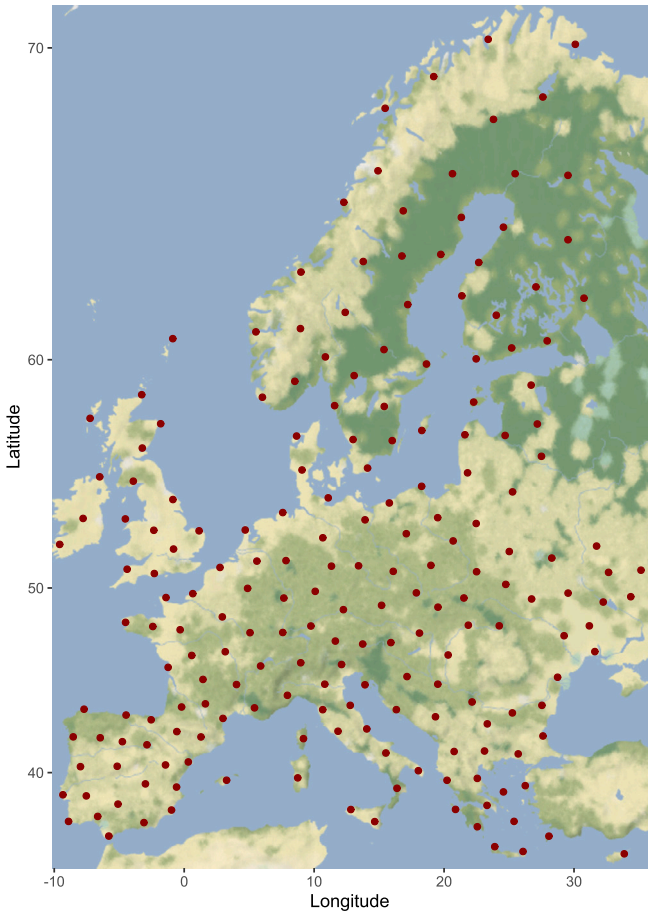

Fig. 5. Sample locations used as background data for calculating the SHAP values.

components would be enough to model the data. The average scaled MASHAP values of SBSS and SNSS, provided in Appendix A in Table A.3 and Table A.4, are spread more evenly for the latent components. SBSS has average scaled MASHAP values from 0.035 to 0.114 and SNSS from 0.029 to 0.127. This indicates that none of the components can be dropped without losing information of the dataset. Also, based on the scaled MASHAP values, it is hard to determine the most interesting components, which makes the interpretation more difficult.

The two most important latent components provided by each of the methods are inspected more closely. The components based on iVAE are plotted in Fig. 6, and the components given by SBSS and SNSS are plotted in Appendix A Figs. A.7 and A.8, respectively. Of components recovered by iVAE, the first latent field shows north-south behavior by having low values in north and higher ones in south, while the second latent field separates the middle Europe by having higher values there and lower ones in south and north. SBSS and SNSS both have one component separating the area from Ukraine to northern Germany (the second component for SBSS and the first for SNSS), while the other ones do not show that clear spatial structures. For SBSS, the first component has slightly lower values in diagonal area from United Kingdom to Greece while the rest of the map has high values, and the second component for SNSS shows any spatial behavior only in area of Portugal and Spain.

We also calculated the scaled MASHAP values for iVAE's unmixing function estimate, where the original observations are treated as input together with the auxiliary variables, and the output is the latent components. By this, we can determine which observed variables contribute the most to each of the latent components. Same background data is used as provided in Fig. 5. By looking at the scaled MASHAP values for the unmixing function estimate, provided in Table A.5, it is evident that Ba and V contribute the most to the latent component 1, while the latent component 2 has highest contributions of Ca, Sr and Mg.

Finally, we study if the prediction power can be increased by predicting the latent components rather than the original observations. We perform 10-fold crossvalidation by dividing the data randomly to 10 equally sized sets which are used one by one as a test data, while the rest of the data is used as a training data. The goal is to predict the observed chemical elements to the locations of the test data. As the data are multivariate and the variables are dependent on each other, the options are to use some multivariate modeling approach which takes the dependencies into account, or to find independent latent components which can be modeled individually. As a multivariate modeling approach we consider cokriging, which takes cross dependencies of each variable pair into account. Cokriging is compared against latent component approaches, where the latent components are found by iVAE, SBSS and

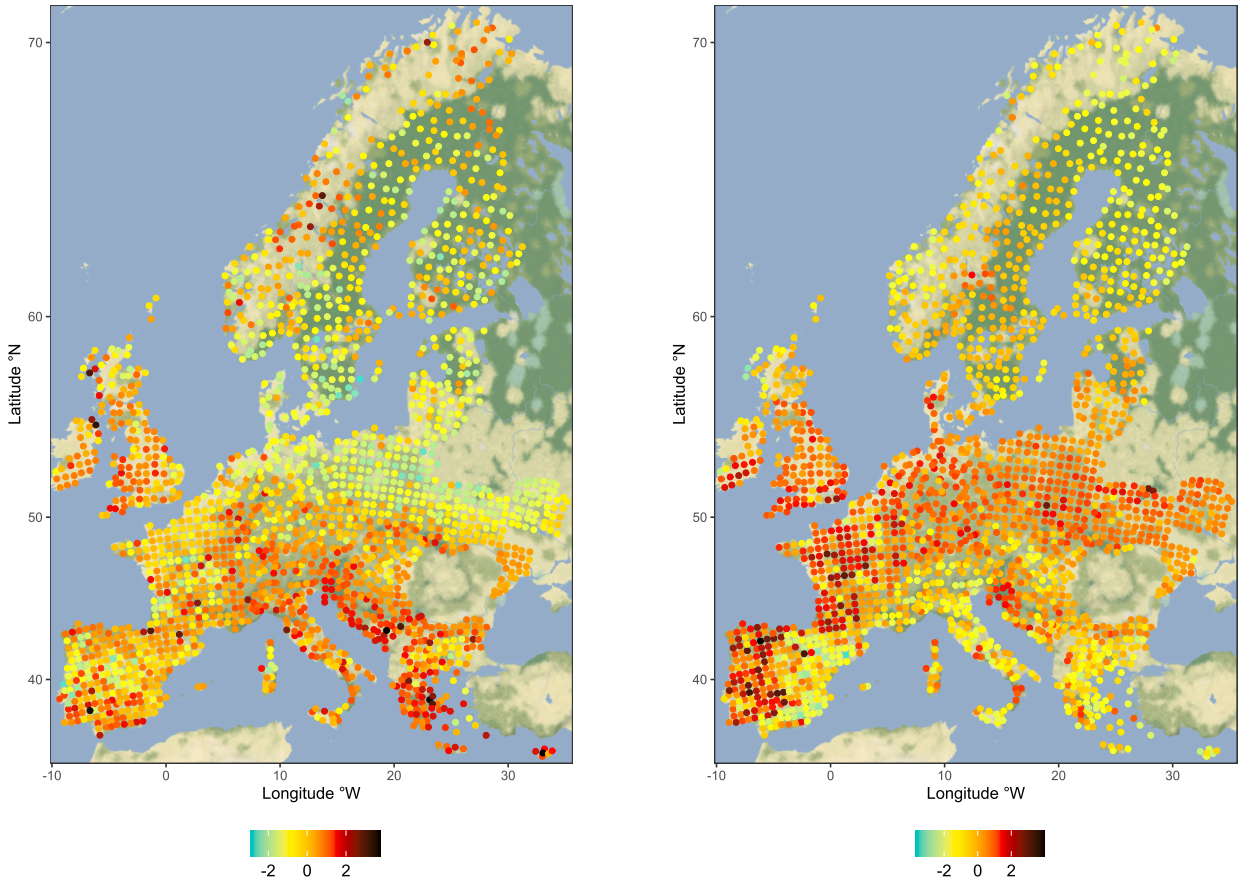

Fig. 6. The first (left) and the second (right) latent fields for GEMAS dataset recovered by iVAE.

SNSS, and the independent latent components are modeled individually using ordinary kriging and universal kriging. Ordinary kriging assumes a stationary mean in the space whereas universal kriging allows a trend in space. For details about kriging and cokriging, see e.g. [32,33].

In cokriging, we fit Matern variogram models for all variograms and cross-variograms of the ilr transformed variables. For variogram models, we select a mutual range parameter by calculating the mean of range parameters of each variable's fitted univariate variograms. The mutual range parameter is selected in order to have a positive definite cokriging system, which is required to calculate predictions to new locations. The Matern parameters for each variogram and cross-variogram are fitted using ordinary least squares (OLS) method. As the predicting with cokriging is computationally very expensive with such many variables and large sample size, we use only 50 nearest points for cokriging predictions to make it feasible.

In ordinary and universal kriging, we fit Matern, exponential and spherical variogram models for each independent component and select the best based on the OLS method. For ordinary kriging, we allow the mean of the spatial field to be a linear function of spatial coordinates. To make a fair comparison, we use only 50 nearest points also for kriging predictions. The predicted latent components are transformed back to ilr variables using each method's mixing function estimate. At last, the predicted data are transformed from ilr space to clr space, where the performance is evaluated. As performance measures, we use mean squared error (MSE), mean absolute error (MAE) and root mean squared error (RMSE). MSE, MAE and RMSE are calculated as $MSE = \frac{1}{n}(x-\hat{x})^2$, $MAE = \frac{1}{n}|(x-\hat{x})|$ and $RMSE = \sqrt{\frac{1}{n}(x-\hat{x})^2}$, where x is the true value and $\hat{x}$ is the estimated one. The average performances over the 10-fold crossvalidation are presented in Table 2. The lowest MSE, MAE and RMSE are obtained when iVAE is combined with ordinary kriging making it the best performing prediction method. SNSS combined with kriging slightly outperforms cokriging, and SBSS combined with kriging has very similar performance as cokriging. Universal kriging does not increase the performance, which indicates that the fields do not have linear trend in space. Because of the performance gain of iVAE combined with kriging, and the fact univariate kriging is computationally much more feasible than cokriging, we consider iVAE combined with kriging the best prediction method.

7. Discussion

Based on the results of this paper, iVAE is a preferable method in settings, where the variances of the latent fields are not stable across space. However, in stationary settings where the sample mean and sample variance did not change enough, SBSS still

Table 2
MSE, MAE and RMSE prediction errors of 10-fold crossvalidation, where the chemical elements of the GEMAS dataset were predicted to new spatial locations using either cokriging or iVAE, SBSS and SNSS combined with ordinary or universal kriging.

Method	MSE	MAE	RMSE
Cokriging	0.1811	0.2787	0.4252
iVAE + Ordinary kriging	**0.1768**	**0.2717**	**0.4201**
iVAE + Universal kriging	0.1770	0.2719	0.4203
SBSS + Ordinary kriging	0.1819	0.2777	0.4261
SBSS + Universal kriging	0.1825	0.2781	0.4268
SNSS + Ordinary kriging	0.1792	0.2758	0.4230
SNSS + Universal kriging	0.1797	0.2762	0.4236

performed better. In practice, this means for example having small range and large shape parameters in a Matern covariance function. However, many real-world spatial phenomena, such as temperature or humidity, often show high spatial dependence making iVAE the preferable method. To overcome the performance drop in stationary settings with low spatial dependence, it requires development of new models/methods which do not assume nonstationarity for identifiability.

In the geochemical application, we calculated the average scaled MASHAP values of the latent components of spatial iVAE's mixing function estimate, and discovered that fewer components might be sufficient to model the dataset. iVAE allows modeling the data with fewer latent components than the observed variables but currently there are, to the best of our knowledge, no methods for estimating the latent dimension.

In our simulation studies, TCL performed poorly in all settings. This is most likely due to nonoptimal simulation setups for the method. TCL relies on classifying observed data into chosen segments; when the classification task is too easy (e.g. we have data with changing mean or the number of real clusters is too low), TCL succeeds in classification without finding ICs. This problem was previously acknowledged also in [13].

8. Conclusions

In this paper we extended iVAE, first proposed by [13], to the nonlinear SNSS setting. We discussed the theoretical background of the method and provided useful tools for interpreting the latent components through nonlinear mixing environment using SHAP values. We used simulation studies to illustrate the performance of iVAE method in nonlinear SBSS and SNSS settings, and applied the method for a geochemical dataset.

Simulation studies revealed that iVAE outperformed its competitors in all settings where the variability in variance of the latent fields was relatively high, that is, the variances changed relatively much between different locations. The variability was achieved either by having multiple (in our simulations, ten) clusters with varying variances, by having latent fields with a nonstationary correlation structure or by having latent fields with stationary correlation structure that yields strong spatial dependence and enough variability in sample variances in space. In the stationary setting, where the mean and variance did not change enough, SBSS still performed better.

In the geochemical application, we utilized iVAE, SBSS and SNSS to find the latent fields for the geochemical dataset. We interpreted the latent fields provided by SBSS, SNSS and iVAE by calculating the scaled MASHAP values for the mixing function estimates, and discovered which fields are the most important ones for each method. Based on the scaled MASHAP values, iVAE provided the easiest interpretable components, two of which were the most important ones. We examined the results further by comparing the two most important component of SBSS, SNSS and iVAE by plotting the corresponding latent fields. iVAE provided clearer structure for the components than the competing methods. In depth interpretations of the latent components, however, requires subject expertise and is beyond the scope of this paper. We also studied if the prediction power can be increased by preprocessing the data using iVAE, and then predicting the independent latent components univariately, and backtransforming the predictions to the original variable space. The results suggest that by using iVAE as pre-processing method, the prediction power can be increased compared to alternative multivariate predicting approach.

In future research, models and methods for nonlinear SBSS should be developed also for stationary data, as iVAE relies on nonstationary variances for identifiability. Furthermore, in our simulations, we only considered Gaussian random fields and assumed Gaussian latent components in model specification. In the future, deeper analysis of iVAE's performance must be done using simulations and various real data examples, where then also the effect of different source density distributions and the robustness properties under outliers or model mismatch must be studied, as was done for SBSS in [34]. The methods should be compared also in more favorable settings for TCL. Considering the flexibility of iVAE in terms of possibilities for auxiliary data, the method could be extended to various situations, such as for spatio-temporal data, as was also recently considered in a linear BSS framework [35]. For the latent field interpretation, we considered MASHAP based procedure for population level interpretations. In future work, other approaches, such as Shapley additive global importance [23] should be explored for latent component interpretation. Finally, in future work, procedures for estimating the dimension of the latent representation will be developed.

CRediT authorship contribution statement

Mika Sipilä: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Klaus Nordhausen:** Writing – review & editing, Supervision, Resources, Project administration, Conceptualization. **Sara Taskinen:** Writing – review & editing, Supervision, Resources, Project administration, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We thank the three anonymous reviewers for helping us improve and clarify this manuscript with their insightful comments. This work was partly supported by the Austrian Science Fund (P31881-N32), the Research Council of Finland (453691), the HiTEc COST Action (CA21163), the Vilho, Yrjö and Kalle Väisälä Foundation, and the Kone foundation (201903741). We also wish to acknowledge CSC – IT Center for Science, Finland, for computational resources.

Appendix A. Additional results for a real data example

Table A.3
Scaled MASHAP values for SBSS method's mixing function estimate.

	IC3	IC2	IC4	IC1	IC5	IC6	IC7	IC12	IC11	IC15	IC8	IC17	IC10	IC9	IC13	IC16	IC14
clr(Al)	0.033	0.157	0.168	0.032	0.125	0.013	0.001	0.034	0.080	0.039	0.113	0.060	0.002	0.050	0.024	0.067	0.002
clr(Ba)	0.193	0.074	0.068	0.063	0.050	0.054	0.013	0.079	0.040	0.033	0.133	0.065	0.028	0.046	0.014	0.013	0.032
clr(Ca)	0.015	0.134	0.186	0.005	0.136	0.016	0.052	0.090	0.055	0.042	0.078	0.020	0.061	0.008	0.057	0.018	0.028
clr(Cr)	0.310	0.016	0.147	0.038	0.069	0.081	0.045	0.031	0.037	0.006	0.007	0.012	0.005	0.044	0.076	0.022	0.053
clr(Fe)	0.129	0.160	0.053	0.037	0.003	0.088	0.033	0.104	0.086	0.015	0.004	0.117	0.020	0.051	0.075	0.002	0.023
clr(K)	0.170	0.047	0.102	0.058	0.089	0.121	0.085	0.015	0.064	0.013	0.084	0.091	0.003	0.010	0.009	0.025	0.013
clr(Mg)	0.116	0.205	0.091	0.089	0.022	0.022	0.061	0.013	0.037	0.037	0.048	0.011	0.080	0.133	0.012	0.014	0.009
clr(Mn)	0.160	0.038	0.046	0.029	0.025	0.015	0.012	0.043	0.011	0.135	0.053	0.131	0.098	0.013	0.049	0.033	0.109
clr(Na)	0.073	0.075	0.054	0.297	0.086	0.103	0.010	0.015	0.026	0.038	0.009	0.050	0.017	0.037	0.004	0.055	0.050
clr(Nb)	0.009	0.052	0.132	0.153	0.016	0.037	0.053	0.061	0.028	0.081	0.065	0.037	0.024	0.079	0.097	0.034	0.041
clr(P)	0.065	0.100	0.094	0.029	0.004	0.111	0.019	0.112	0.035	0.070	0.027	0.029	0.067	0.065	0.049	0.088	0.035
clr(Si)	0.152	0.228	0.003	0.138	0.154	0.027	0.032	0.019	0.021	0.039	0.016	0.018	0.011	0.094	0.030	0.003	0.016
clr(Sr)	0.113	0.088	0.121	0.084	0.106	0.026	0.010	0.056	0.081	0.007	0.009	0.059	0.072	0.034	0.005	0.035	0.095
clr(Ti)	0.050	0.034	0.101	0.089	0.035	0.031	0.193	0.055	0.050	0.032	0.027	0.019	0.102	0.016	0.082	0.054	0.030
clr(V)	0.182	0.085	0.025	0.012	0.085	0.143	0.059	0.084	0.103	0.041	0.008	0.001	0.084	0.002	0.007	0.029	0.050
clr(Y)	0.002	0.042	0.083	0.189	0.019	0.085	0.102	0.063	0.023	0.068	0.025	0.035	0.063	0.036	0.059	0.073	0.034
clr(Zn)	0.146	0.003	0.163	0.091	0.025	0.075	0.137	0.068	0.073	0.051	0.042	0.011	0.025	0.013	0.020	0.049	0.008
clr(Zr)	0.131	0.245	0.032	0.164	0.038	0.031	0.121	0.036	0.005	0.076	0.045	0.006	0.001	0.005	0.003	0.057	0.003
Average	0.114	0.099	0.093	0.089	0.060	0.060	0.058	0.054	0.047	0.046	0.044	0.043	0.042	0.041	0.037	0.037	0.035

Table A.4
Scaled MASHAP values for SNSS method's mixing function estimate.

	IC2	IC5	IC7	IC3	IC4	IC6	IC8	IC1	IC14	IC9	IC10	IC16	IC12	IC17	IC11	IC15	IC13
clr(Al)	0.096	0.216	0.017	0.003	0.015	0.063	0.007	0.165	0.040	0.068	0.079	0.062	0.031	0.066	0.011	0.011	0.051
clr(Ba)	0.150	0.095	0.034	0.026	0.093	0.027	0.046	0.022	0.053	0.030	0.134	0.037	0.088	0.064	0.061	0.010	0.029
clr(Ca)	0.102	0.263	0.026	0.041	0.046	0.032	0.007	0.068	0.091	0.067	0.019	0.030	0.025	0.077	0.009	0.064	0.035
clr(Cr)	0.131	0.037	0.068	0.019	0.186	0.128	0.043	0.089	0.059	0.035	0.064	0.002	0.027	0.037	0.004	0.041	0.030
clr(Fe)	0.196	0.046	0.027	0.030	0.062	0.056	0.018	0.056	0.139	0.023	0.082	0.002	0.055	0.049	0.014	0.042	0.104
clr(K)	0.112	0.107	0.024	0.113	0.072	0.022	0.061	0.060	0.037	0.073	0.073	0.016	0.076	0.063	0.021	0.041	0.028
clr(Mg)	0.238	0.111	0.029	0.053	0.021	0.002	0.157	0.073	0.026	0.021	0.114	0.016	0.046	0.001	0.008	0.039	0.045
clr(Mn)	0.070	0.034	0.040	0.047	0.041	0.071	0.056	0.099	0.165	0.029	0.013	0.054	0.110	0.063	0.030	0.062	0.016
clr(Na)	0.005	0.144	0.089	0.124	0.137	0.086	0.063	0.080	0.062	0.014	0.021	0.099	0.008	0.017	0.021	0.019	0.012
clr(Nb)	0.078	0.082	0.100	0.099	0.011	0.089	0.099	0.001	0.017	0.080	0.004	0.070	0.067	0.012	0.116	0.053	0.021
clr(P)	0.113	0.057	0.083	0.029	0.028	0.058	0.004	0.081	0.085	0.145	0.000	0.169	0.023	0.039	0.049	0.028	0.008
clr(Si)	0.289	0.082	0.044	0.069	0.043	0.082	0.057	0.003	0.030	0.119	0.025	0.011	0.064	0.017	0.045	0.019	0.001

Table A.4 (*continued*)

	IC2	IC5	IC7	IC3	IC4	IC6	IC8	IC1	IC14	IC9	IC10	IC16	IC12	IC17	IC11	IC15	IC13
clr(Sr)	0.003	0.152	0.044	0.112	0.093	0.031	0.040	0.064	0.032	0.080	0.084	0.045	0.081	0.058	0.071	0.004	0.008
clr(Ti)	0.050	0.106	0.131	0.001	0.053	0.070	0.114	0.014	0.089	0.031	0.133	0.073	0.027	0.066	0.031	0.005	0.005
clr(V)	0.153	0.016	0.068	0.047	0.072	0.097	0.075	0.030	0.057	0.033	0.089	0.006	0.000	0.056	0.063	0.065	0.072
clr(Y)	0.079	0.071	0.151	0.191	0.018	0.000	0.106	0.030	0.006	0.042	0.018	0.011	0.055	0.049	0.048	0.094	0.028
clr(Zn)	0.086	0.031	0.154	0.164	0.126	0.157	0.064	0.059	0.051	0.053	0.003	0.009	0.003	0.015	0.018	0.003	0.004
clr(Zr)	0.328	0.063	0.137	0.092	0.028	0.000	0.050	0.063	0.001	0.039	0.017	0.092	0.006	0.026	0.031	0.008	0.019
Average	0.127	0.095	0.070	0.070	0.064	0.060	0.059	0.059	0.058	0.055	0.054	0.045	0.044	0.043	0.036	0.034	0.029

Table A.5
The scaled MASHAP values for the iVAE model's encoder part.

	IC1	IC2	IC3	IC4	IC5	IC6	IC7	IC8	IC9	IC10	IC11	IC12	IC13	IC14	IC15	IC16	IC17
clr(Al)	0.042	0.027	0.024	0.014	0.042	0.039	0.050	0.011	0.109	0.017	0.008	0.082	0.009	0.036	0.038	0.042	0.036
clr(Ba)	0.128	0.047	0.017	0.029	0.086	0.009	0.142	0.037	0.201	0.018	0.060	0.095	0.059	0.060	0.059	0.069	0.057
clr(Ca)	0.057	0.221	0.028	0.089	0.039	0.155	0.114	0.042	0.063	0.090	0.133	0.028	0.038	0.066	0.075	0.070	0.085
clr(Cr)	0.029	0.031	0.013	0.084	0.014	0.063	0.028	0.155	0.031	0.134	0.027	0.027	0.062	0.058	0.053	0.065	0.058
clr(Fe)	0.055	0.015	0.028	0.043	0.033	0.074	0.056	0.019	0.019	0.013	0.089	0.128	0.014	0.039	0.045	0.053	0.048
clr(K)	0.062	0.006	0.062	0.103	0.031	0.008	0.007	0.059	0.015	0.070	0.023	0.087	0.015	0.060	0.059	0.066	0.052
clr(Mg)	0.072	0.110	0.161	0.068	0.129	0.086	0.015	0.035	0.019	0.046	0.029	0.025	0.171	0.073	0.097	0.064	0.068
clr(Mn)	0.042	0.024	0.125	0.006	0.034	0.098	0.215	0.027	0.058	0.008	0.025	0.045	0.032	0.058	0.054	0.051	0.051
clr(Na)	0.057	0.086	0.043	0.057	0.156	0.069	0.038	0.017	0.038	0.079	0.101	0.022	0.076	0.068	0.053	0.062	0.057
clr(Nb)	0.015	0.052	0.019	0.026	0.027	0.016	0.007	0.074	0.050	0.086	0.036	0.130	0.024	0.052	0.053	0.050	0.069
clr(P)	0.044	0.009	0.134	0.072	0.067	0.087	0.192	0.051	0.007	0.020	0.031	0.032	0.015	0.053	0.051	0.053	0.047
clr(Si)	0.078	0.033	0.007	0.016	0.057	0.063	0.011	0.141	0.101	0.042	0.071	0.011	0.086	0.066	0.050	0.034	0.064
clr(Sr)	0.051	0.102	0.068	0.035	0.057	0.010	0.017	0.068	0.032	0.071	0.160	0.008	0.034	0.047	0.049	0.057	0.054
clr(Ti)	0.028	0.054	0.043	0.046	0.055	0.012	0.026	0.087	0.044	0.044	0.074	0.007	0.051	0.056	0.056	0.065	0.063
clr(V)	0.093	0.018	0.019	0.054	0.057	0.038	0.011	0.035	0.095	0.151	0.020	0.017	0.057	0.048	0.050	0.051	0.043
clr(Y)	0.012	0.055	0.067	0.030	0.087	0.034	0.019	0.061	0.104	0.017	0.029	0.072	0.136	0.054	0.046	0.048	0.058
clr(Zn)	0.059	0.039	0.112	0.166	0.012	0.115	0.042	0.063	0.007	0.064	0.026	0.048	0.011	0.052	0.050	0.051	0.040
clr(Zr)	0.077	0.070	0.032	0.064	0.018	0.026	0.009	0.017	0.007	0.030	0.060	0.134	0.110	0.053	0.063	0.049	0.050

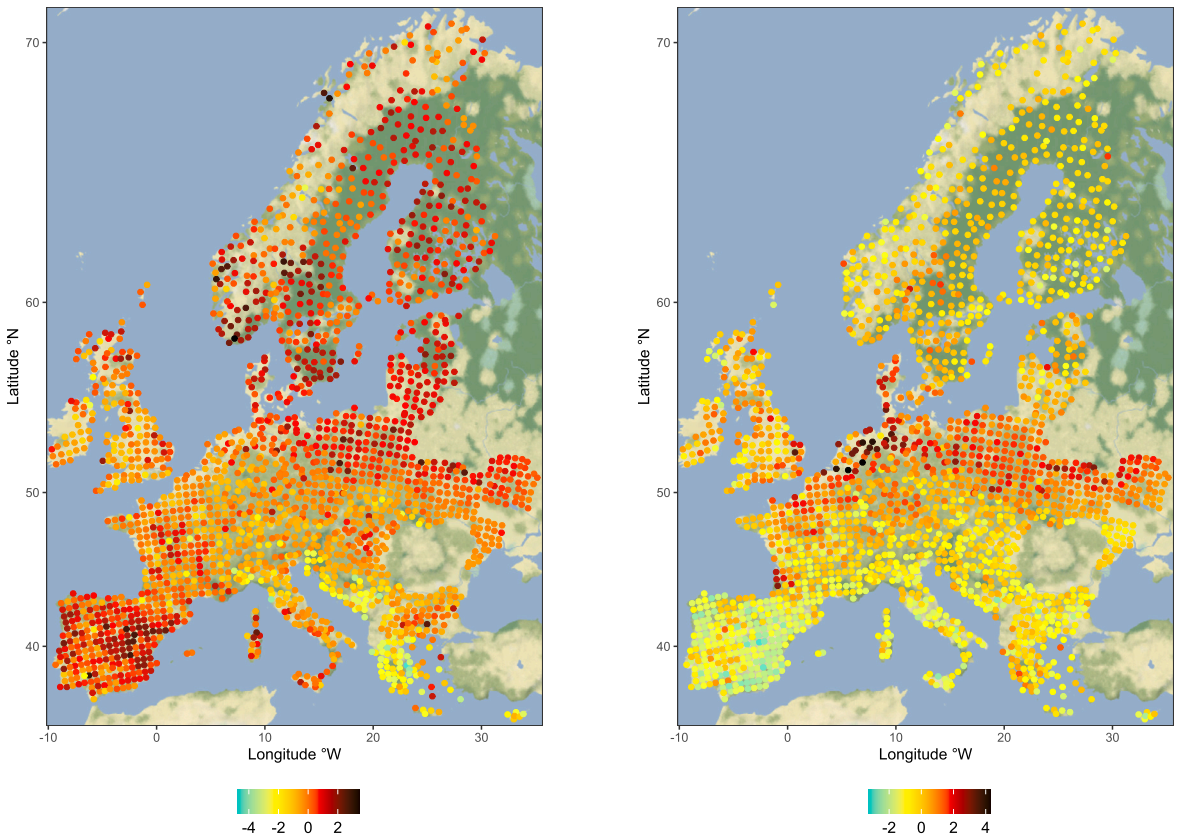

Fig. A.7. The first (left) and the second (right) latent fields for GEMAS dataset recovered by SBSS.

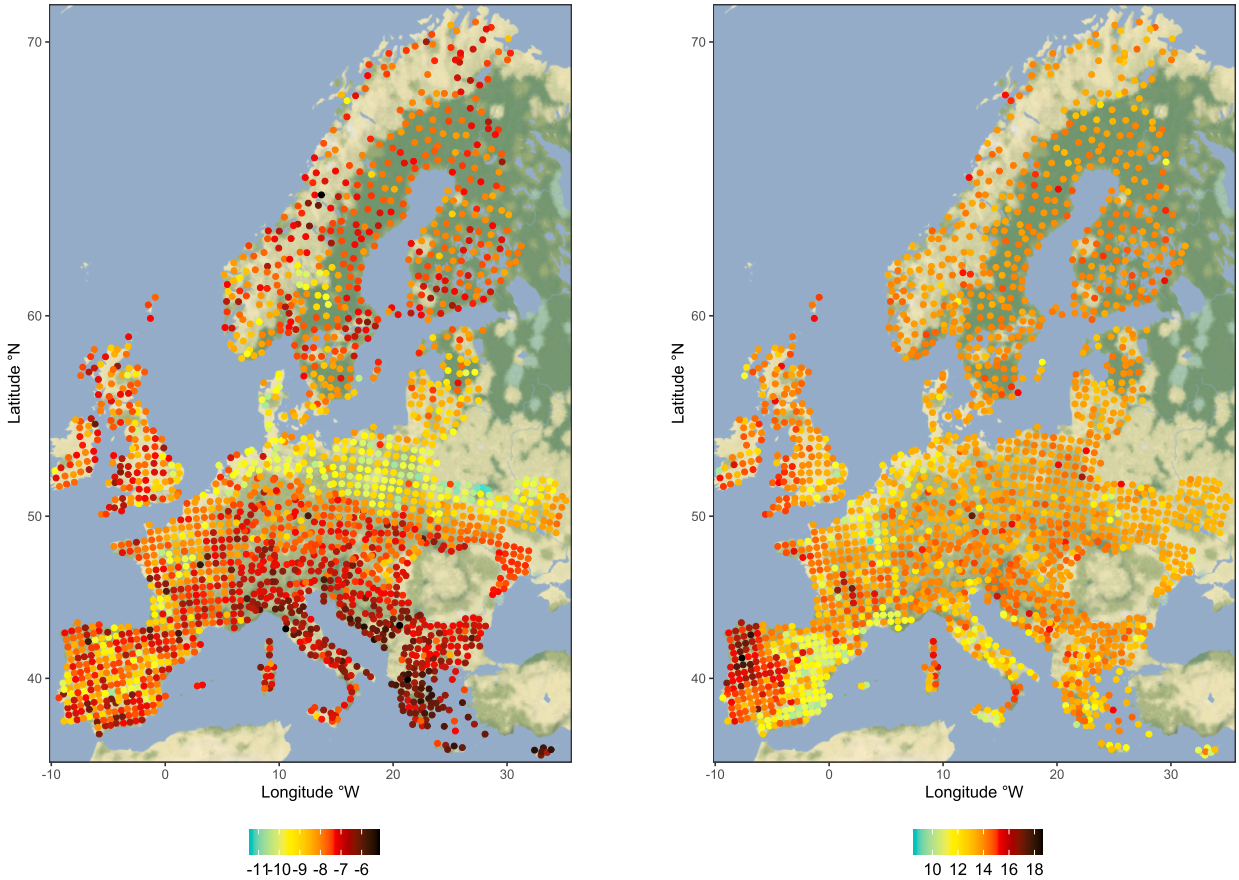

Fig. A.8. The first (left) and the second (right) latent fields for GEMAS dataset recovered by SNSS.

References

[1] M.G. Genton, W. Kleiber, Cross-covariance functions for multivariate geostatistics, Stat. Sci. 30 (2015) 147–163, https://doi.org/10.1214/14-STS487.

[2] C. Muehlmann, K. Nordhausen, M. Yi, On cokriging, neural networks, and spatial blind source separation for multivariate spatial prediction, IEEE Geosci. Remote Sens. Lett. 18 (11) (2021) 1931–1935, https://doi.org/10.1109/LGRS.2020.3011549.

[3] P. Comon, C. Jutten, Handbook of Blind Source Separation: Independent Component Analysis and Applications, Academic Press, 2010.

[4] K. Nordhausen, H. Oja, P. Filzmoser, C. Reimann, Blind source separation for spatial compositional data, Math. Geosci. 47 (7) (2015) 753–770, https://doi.org/10.1007/s11004-014-9559-5.

[5] F. Bachoc, M.G. Genton, K. Nordhausen, A. Ruiz-Gazen, J. Virta, Spatial blind source separation, Biometrika 107 (2020) 627–646, https://doi.org/10.1093/biomet/asz079.

[6] C. Muehlmann, F. Bachoc, K. Nordhausen, M. Yi, Test of the latent dimension of a spatial blind source separation model, in press Stat. Sin. (2024), https://doi.org/10.5705/ss.202021.0326.

[7] C. Muehlmann, F. Bachoc, K. Nordhausen, Blind source separation for non-stationary random fields, Spat. Stat. 47 (2022) 100574, https://doi.org/10.1016/j.spasta.2021.100574.

[8] A. Hyvärinen, I. Khemakhem, H. Morioka, Nonlinear independent component analysis for principled disentanglement in unsupervised deep learning, Patterns 4 (10) (2023) 100844, https://doi.org/10.1016/j.patter.2023.100844, https://www.sciencedirect.com/science/article/pii/S2666389923002234.

[9] A. Hyvärinen, P. Pajunen, Nonlinear independent component analysis: existence and uniqueness results, Neural Netw. 12 (3) (1999) 429–439, https://doi.org/10.1016/S0893-6080(98)00140-3.

[10] D.P. Kingma, M. Welling, Auto-encoding variational Bayes, arXiv preprint, arXiv:1312.6114, 2013, https://doi.org/10.48550/arXiv.1312.6114.

[11] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, A. Lerchner, β-VAE: learning basic visual concepts with a constrained variational framework, in: International Conference on Learning Representations, 2017.

[12] S. Zhao, J. Song, S. Ermon, InfoVAE: information maximizing variational autoencoders, arXiv preprint, arXiv:1706.02262, 2017, https://doi.org/10.48550/arXiv.1706.02262.

[13] I. Khemakhem, D. Kingma, R. Monti, A. Hyvärinen, Variational autoencoders and nonlinear ICA: a unifying framework, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2020, pp. 2207–2217.

[14] H. Hälvä, S. Le Corff, L. Lehéricy, J. So, Y. Zhu, E. Gassiat, A. Hyvärinen, Disentangling identifiable features from noisy data with structured nonlinear ICA, Adv. Neural Inf. Process. Syst. 34 (2021) 1624–1633.

[15] S.M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, Adv. Neural Inf. Process. Syst. 30 (2017), https://doi.org/10.48550/arXiv.1705.07874.

[16] A. Hyvärinen, H. Morioka, Unsupervised feature extraction by time-contrastive learning and nonlinear ICA, Adv. Neural Inf. Process. Syst. 29 (2016), https://doi.org/10.48550/arXiv.1605.06336.

[17] A. Hyvärinen, H. Morioka, Nonlinear ICA of temporally dependent stationary sources, in: Artificial Intelligence and Statistics, PMLR, 2017, pp. 460–469.

[18] A. Hyvärinen, H. Sasaki, R. Turner, Nonlinear ICA using auxiliary variables and generalized contrastive learning, in: The 22nd International Conference on Artificial Intelligence and Statistics, PMLR, 2019, pp. 859–868.
[19] H. Hälvä, A. Hyvärinen, Hidden Markov nonlinear ICA: unsupervised learning from nonstationary time series, in: Conference on Uncertainty in Artificial Intelligence, PMLR, 2020, pp. 939–948.
[20] L.S. Shapley, 17. A value for n-person games, Princeton University Press, Princeton, 1953, pp. 307–318.
[21] M.T. Ribeiro, S. Singh, C. Guestrin, "Why should I trust you?": explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 1135–1144.
[22] W.E. Marcílio, D.M. Eler, From explanations to feature selection: assessing SHAP values as feature selection mechanism, in: 2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), IEEE, 2020, pp. 340–347.
[23] I. Covert, S.M. Lundberg, S.-I. Lee, Understanding global feature contributions with additive importance measures, Adv. Neural Inf. Process. Syst. 33 (2020) 17212–17223.
[24] R Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, 2023, https://www.R-project.org/.
[25] C. Muehlmann, M. Sipilä, K. Nordhausen, S. Taskinen, J. Virta, SpatialBSS: Blind Source Separation for Multivariate Spatial Data, R package version 0.13-0, https://CRAN.R-project.org/package=SpatialBSS, 2022.
[26] E.J. Pebesma, Multivariable geostatistics in S: the gstat package, Comput. Geosci. 30 (2004) 683–691.
[27] E.B. Anderes, M.L. Stein, Local likelihood estimation for nonstationary random fields, J. Multivar. Anal. 102 (3) (2011) 506–520, https://doi.org/10.1016/j.jmva.2010.10.010.
[28] D.-A. Clevert, T. Unterthiner, S. Hochreiter, Fast and accurate deep network learning by exponential linear units (ELUs), arXiv preprint, arXiv:1511.07289, 2015, https://doi.org/10.48550/arXiv.1511.07289.
[29] C. Reimann, M. Birke, A. Demetriades, P. Filzmoser, P. O'Connor, Chemistry of Europe's Agricultural Soils, Part A, Schweizerbart'sche Verlagsbuchhandlung, Stuttgart, Germany, 2014.
[30] P. Filzmoser, K. Hron, M. Templ, Applied Compositional Data Analysis, Springer, Cham, 2018.
[31] J.J. Egozcue, V. Pawlowsky-Glahn, G. Mateu-Figueras, C. Barcelo-Vidal, Isometric logratio transformations for compositional data analysis, Math. Geol. 35 (3) (2003) 279–300, https://doi.org/10.1023/A:1023818214614.
[32] J.-P. Chilès, N. Desassis, Fifty years of kriging, in: Handbook of Mathematical Geosciences: Fifty Years of IAMG, 2018, pp. 589–612.
[33] P. Goovaerts, Ordinary cokriging revisited, Math. Geol. 30 (1998) 21–42.
[34] M. Sipilä, C. Muehlmann, K. Nordhausen, S. Taskinen, Robust second-order stationary spatial blind source separation using generalized sign matrices, Spat. Stat. 59 (2024) 100803, https://doi.org/10.1016/j.spasta.2023.100803.
[35] C. Muehlmann, S. De Iaco, K. Nordhausen, Blind recovery of sources for multivariate space-time random fields, Stoch. Environ. Res. Risk Assess. 37 (2023) 1593–1613, https://doi.org/10.1007/s00477-022-02348-2.