

JYX



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; Cappello, Claudia; De Iaco, Sandra; Nordhausen, Klaus; Taskinen, Sara

Title: Modelling multivariate spatio-temporal data with identifiable variational autoencoders

Year: 2025

Version: Published version

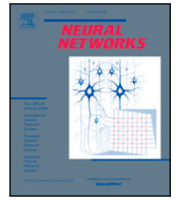
Copyright: © 2024 The Authors. Published by Elsevier Ltd.

Rights: CC BY 4.0

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

Please cite the original version:

Sipilä, M., Cappello, C., De Iaco, S., Nordhausen, K., & Taskinen, S. (2025). Modelling multivariate spatio-temporal data with identifiable variational autoencoders. *Neural Networks*, 181, Article 106774. <https://doi.org/10.1016/j.neunet.2024.106774>



Full Length Article



Modelling multivariate spatio-temporal data with identifiable variational autoencoders

Mika Sipilä^{a,*}, Claudia Cappello^b, Sandra De Iaco^b, Klaus Nordhausen^a, Sara Taskinen^a^a Department of Mathematics and Statistics, University of Jyväskylä, Finland^b DSE - Section of Mathematics and Statistics, University of Salento, Italy

ARTICLE INFO

Keywords:

Blind source separation
Dimension estimation
Kriging
Meteorological data
Shapley values

ABSTRACT

Modelling multivariate spatio-temporal data with complex dependency structures is a challenging task but can be simplified by assuming that the original variables are generated from independent latent components. If these components are found, they can be modelled univariately. Blind source separation aims to recover the latent components by estimating the unknown linear or nonlinear unmixing transformation based on the observed data only. In this paper, we extend recently introduced identifiable variational autoencoder to the nonlinear nonstationary spatio-temporal blind source separation setting and demonstrate its performance using comprehensive simulation studies. Additionally, we introduce two alternative methods for the latent dimension estimation, which is a crucial task in order to obtain the correct latent representation. Finally, we illustrate the proposed methods using a meteorological application, where we estimate the latent dimension and the latent components, interpret the components, and show how nonstationarity can be accounted and prediction accuracy can be improved by using the proposed nonlinear blind source separation method as a preprocessing method.

1. Introduction

Many real world phenomena, such as weather, epidemiological patterns and ecosystem dynamics, are multivariate spatio-temporal, meaning that multivariate observations $\mathbf{x}(s, t) := \mathbf{x} \in \mathbb{R}^S$ are observed in a spatial location $s \in S \subset \mathbb{R}^D$ at time $t \in \mathcal{T} \subset \mathbb{R}$, where S is called a spatial domain, \mathcal{T} is called a temporal domain and D is a spatial dimension. Without loss of generality, we assume from now on that $D = 2$. A multivariate observation \mathbf{x} contains measurements of multiple, usually dependent, random variables describing the phenomenon of interest. When modelling such multivariate spatio-temporal data, one has to account not only the dependence between the variables, but also the dependences in space and in time. The dependence structure is often described through spatio-temporal covariance function $C(\mathbf{x}(s, t), \mathbf{x}(s', t'))$, where (s, t) and (s', t') are two spatio-temporal locations. The covariance C is a $S \times S$ matrix valued functional with elements C_{ij} , $i, j = 1, \dots, S$, defined as $C_{ij} = C(x_i(s, t), x_j(s', t')) = E(x_i(s, t)x_j(s', t')) - E(x_i(s, t))E(x_j(s', t'))$. Modelling the covariance function C is usually a highly demanding task, and often, in order to make the modelling feasible, some severely restricting assumptions, such as stationarity or separability, are made. When the spatio-temporal field is assumed to be stationary, the covariance function can be simplified

to

$$C(\mathbf{x}(s, t), \mathbf{x}(s', t')) = C(\|s - s'\|, |t - t'|), \quad (1)$$

meaning that the value of the function depends only on the distance between the spatial locations and the distance between temporal locations. If (1) does not hold, the data are nonstationary, meaning that the covariance function may differ when spatial or temporal location is altered. When separability is assumed, the spatio-temporal covariance function can be written as a product of spatial and temporal covariance functions as

$$C(\mathbf{x}(s, t), \mathbf{x}(s', t')) = C_S(s, s')C_T(t, t'),$$

meaning that the spatial and temporal covariance models can be fitted independently and that the spatio-temporal interaction is not considered. The assumptions of stationarity or separability often lead to unrealistically simple models that hence produce nonoptimal results under nonstationary or nonseparable data. Accounting complex nonseparable and nonstationary correlation structures is complicated already in the univariate case, for which an overview can be found in Chen, Genton, and Sun (2021). For multivariate data, the task is even more demanding and computationally challenging as the cross-dependencies between the variables have to be taken into account.

* Corresponding author.

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

For more details of complexity of nonstationary covariance functions for multivariate spatio-temporal data, see Porcu, Furrer, and Nychka (2021), Salvana and Genton (2020).

Another approach to simplify the modelling is to assume that the observations are composed of P latent, mutually independent components (ICs) $\mathbf{z}(s, t) := \mathbf{z} \in \mathbb{R}^P$ through some mixing environment. The main motivation for assuming the ICs is, that if the latent components \mathbf{z} are recovered, they can be modelled univariately making for example nonstationarity much easier to account for. Being able to model components univariately is especially desirable in spatio-temporal settings, where multivariate modelling is highly demanding and computationally challenging as discussed previously. Additionally, the ICs may reveal some meaningful patterns and structures in the observed data that can lead to new insights of the phenomenon of interest. A linear blind source separation (BSS) (Comon & Jutten, 2010) is a popular approach to recover the latent components \mathbf{z} . In linear BSS, it is assumed that the mixing environment is linear and usually also that $S = P$ meaning that a P -variate observable random vector $\mathbf{x} = (x_1, \dots, x_P)^\top$ is generated as

$$\mathbf{x} = \mathbf{A}\mathbf{z}, \quad (2)$$

where \mathbf{A} is an invertible $P \times P$ mixing matrix and $\mathbf{z} = (z_1, \dots, z_P)^\top$ are the P -variate latent components. The objective is to recover \mathbf{A} and \mathbf{z} using only \mathbf{x} and varying assumptions on \mathbf{z} depending on the method used. For example, spatial BSS (SBSS) (Bachoc, Genton, Nordhausen, Ruiz-Gazen, & Virta, 2020; Nordhausen, Oja, Filzmoser, & Reimann, 2015), which is a method for multivariate stationary spatial data, assumes spatially stationary \mathbf{z} , and a nonstationary extension of SBSS, spatial nonstationary source separation (SNSS) (Muehlmann, Bachoc, & Nordhausen, 2022), assumes \mathbf{z} to have nonstationary spatial covariance function. SBSS and SNSS recover the latent components by jointly diagonalizing two or more moment-based matrices. Recently, SBSS was also extended for stationary spatio-temporal data yielding spatio-temporal BSS (STBSS) (Muehlmann, De Iaco, & Nordhausen, 2023). A drawback of STBSS and linear BSS methods in general is that they assume linear mixing (2) which may be too restrictive assumption for many real life applications. Similarly, the assumption that there are as many latent ‘‘signal’’ components as observed variables is in many applications undesirable and it is often hoped that there are significantly fewer signals. This assumption is often needed simply due to the lack of tools for estimating the correct number of signals. Finally, STBSS is developed only for stationary data, and to our knowledge, there are no spatio-temporal alternatives available for nonstationary data cases.

Recent advancements in unsupervised deep learning, such as variational autoencoders (VAEs) (Kingma & Welling, 2013) and generative adversarial networks (GANs) (Goodfellow et al., 2020), have increased interest for developing nonlinear BSS methods, where the mixing function is not restricted to be linear, but can be any injective function $f : \mathbb{R}^P \rightarrow \mathbb{R}^S$, which generates the observed data \mathbf{x} as

$$\mathbf{x} = f(\mathbf{z}). \quad (3)$$

The objective is then to identify an unmixing transformation $q : \mathbb{R}^S \rightarrow \mathbb{R}^P$, which returns the latent components \mathbf{z} as

$$\mathbf{z} = q(\mathbf{x})$$

based on the observations \mathbf{x} only. Without any additional assumptions on the mixing transformations f or on the latent components \mathbf{z} , the model is unidentifiable as there exists infinite nonlinear transformations to generate mutually independent components from the observations (Hyvärinen & Pajunen, 1999). For this reason both VAEs and GANs, in general, suffer from the unidentifiability issue. However, in many recent studies (Hälvä & Hyvärinen, 2020; Hyvärinen & Morioka, 2016, 2017; Hyvärinen, Sasaki, & Turner, 2019; Khemakhem, Kingma, Monti, & Hyvärinen, 2020) the identifiability have been achieved by

introducing some constraints on the distribution of the latent components \mathbf{z} . The main assumption leading to identifiability is that the components z_1, \dots, z_P are statistically dependent on a m -dimensional auxiliary variable \mathbf{u} , and that the components are conditionally independent yielding the joint distribution $p(\mathbf{z}|\mathbf{u}) = \prod_{i=1}^P p(z_i|\mathbf{u})$. In previous studies, the main focus has been in time series data for which several algorithms and examples for auxiliary variables exist in the literature. In case of stationary time series data, permutation contrastive learning (PCL) (Hyvärinen & Morioka, 2017) can be used, for which \mathbf{u} is usually given by one or more previous observations in time. For nonstationary time series data, the available methods are time contrastive learning (TCL) (Hyvärinen & Morioka, 2016), hidden Markov nonlinear ICA (HM-NICA) (Hälvä & Hyvärinen, 2020) and temporal identifiable VAE (iVAE) (Khemakhem et al., 2020), all of which use the time segment of the observation as \mathbf{u} . Generalized contrastive learning (Hyvärinen et al., 2019) and nonlinear ICA with switching linear dynamical systems (Δ -SNICA) (Hälvä, Le Corff, Lehericy, So, Zhu, Gassiat, & Hyvärinen, 2021) can account both stationary and nonstationary time series. In HM-NICA and Δ -SNICA, the auxiliary variables \mathbf{u} are not explicitly provided by the user, but they are instead assumed to be hidden states that are modelled simultaneously by the algorithms. In Sipilä, Nordhausen and Taskinen (2024), iVAE was studied further and extended to nonstationary spatial setting, where spatial segmentation was used as \mathbf{u} . Hälvä et al. (2021) also introduced a structured nonlinear ICA framework which could be used for spatial process, but did not provide any algorithm for the method. In addition to these more general identifiable nonlinear BSS methods, many other deep learning based BSS methods (Ansari, Alatrany, Alnajjar, Khater, Mahmoud, Al-Jumeily, & Hussain, 2023) have been introduced for mainly acoustic signal specific settings, where only serial dependence is present. However, the spatio-temporal data as discussed in this paper is special in sense that in the temporal domain there is natural direction of dependence (past-future) while in the spatial domain such direction is missing and the dependence is usually considered as a function of the distance between two points. Hence, none of the previous methods are directly applicable or optimal for such spatio-temporal data. Note that regularly spaced spatio-temporal data is often represented as tensor data, and BSS methods developed for such specific cases, like those in Virta and Nordhausen (2017), are generally not applicable to broader spatio-temporal settings.

In particular, we are interested in iVAE, which utilizes the auxiliary variable to make VAE identifiable. iVAE is capable of estimating nonlinear injective mixing function, meaning that it allows the latent dimension P to be less or equal to the observed dimension S . However, the latent dimension P has to be estimated beforehand, and currently the nonlinear BSS framework lacks methods for the latent dimension estimation.

In this paper, iVAE is extended to nonstationary spatio-temporal setting by introducing three novel approaches to construct the auxiliary variables. The proposed methods address two key limitations of previous STBSS approaches: they accommodate nonlinear mixing functions and allow for more observed variables than latent components. Moreover, the developed methods are suitable for nonstationary data, unlike earlier STBSS methods, which rely on the assumption of stationarity. The three developed methods, coordinate based, segmentation based and radial basis function based iVAE algorithms, are studied using comprehensive simulation studies to find how various types of nonstationarity affect the performance of the methods. The best performing method, radial basis function based iVAE, is illustrated in real life meteorological application where the recovered latent components are interpreted, and a new procedure to account for nonstationarity in modelling and predicting multivariate data is demonstrated. Moreover, nonlinear BSS framework currently lacks methods for estimating the latent dimension P , which is a crucial task in order to recover the true latent components and to obtain as low dimensional representation of the data as possible without losing much information. Therefore, two alternative procedures for latent dimension estimation are introduced. To conclude, the main contributions of this paper are:

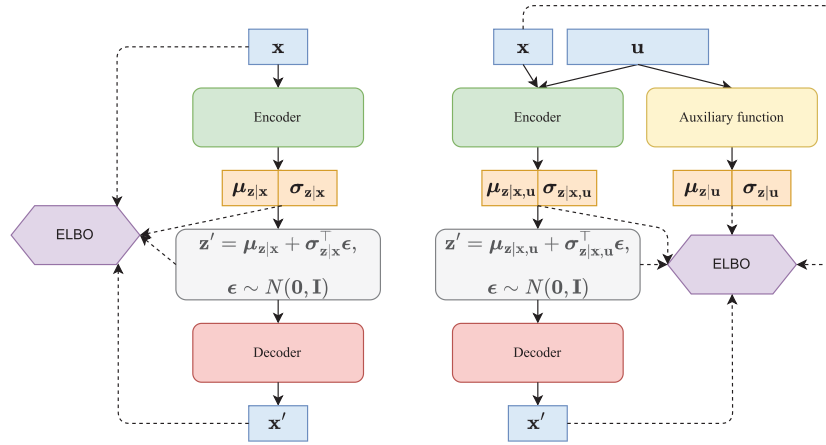


Fig. 1. Schematic representations of VAE (left) and iVAE (right) models. For VAE, the lower bound of the data log likelihood (ELBO) is formed of x , x' , z' , $\mu_{z|x}$ and $\sigma_{z|x}$. In iVAE, ELBO has in addition $\mu_{z|u}$ and $\sigma_{z|u}$ which are provided by the auxiliary function. The latent components are obtained as $\mu_{z|x,u}$.

1. Extending iVAE to the nonstationary spatio-temporal setting by proposing three novel approaches for constructing auxiliary variables.
2. Introducing two alternative procedures for latent dimension estimation.
3. Developing a new iVAE-based method for addressing nonstationarity in the modelling and prediction of spatio-temporal data.

The rest of the paper is organized as follows. In Section 2 we review basic theory behind VAE and iVAE, and discuss the identifiability, after which the spatio-temporal iVAE extensions are introduced in Section 3. In Section 4, the introduced methods are compared using simulation studies, and two alternative latent dimension estimation methods are studied. Finally, Section 5 shows a real data example and Section 6 concludes the paper.

2. Variational autoencoders and identifiability

Let $x \in \mathbb{R}^S$ be an observable random vector and $z \in \mathbb{R}^P$, $P \leq S$, be a latent random vector, *i.e.*, a source vector. Variational autoencoders (VAE) (Kingma & Welling, 2013) assume that the observed data are generated from a deep latent variable model with the structure

$$p^*(x, z) = p^*(x|z)p^*(z),$$

where p^* is a true, unknown generative distribution, $z \sim p^*(z)$ and $x \sim p^*(x|z)$. The distribution of the observed data is then obtained as

$$p^*(x) = \int p^*(x, z) dz.$$

VAE consists of an encoder $g(x)$ and a decoder $h(z)$, which are parameterized by deep neural networks with parameters $\theta = (\theta_g^T, \theta_h^T)^T$. The encoder maps the observed data to mean vector $\mu_{z|x} \in \mathbb{R}^P$ and variance vector $\sigma_{z|x} \in \mathbb{R}^P$, which are used to sample a new latent representation z' by applying the reparametrization trick (Kingma & Welling, 2013). The decoder transforms the latent representation z' back to the observable data x' . The VAE framework allows effective optimization of the parameters θ so that after optimization we have that

$$p_\theta(x) \approx p^*(x).$$

The VAE framework learns the full generative model $p_\theta(x, z) = p_\theta(x|z)p_\theta(z)$ and a variational approximation $q_\theta(z|x)$ of the posterior distribution $p_\theta(z|x)$ by maximizing the lower bound of the data log-likelihood, or evidence lower bound (ELBO), defined as

$$\mathcal{L}(\theta|x) \geq E_{q_\theta(z|x)}(\log p_\theta(x|z) + \log p_\theta(z) - \log q_\theta(z|x))$$

with respect to the parameter vector θ . The problem however is that the model is not identifiable, meaning that even though we have a good estimate of the marginal distribution $p^*(x)$, there is no guarantee that $p_\theta(x, z) \approx p^*(x, z)$. More formally, the model is identifiable if for all (x, z) it holds that

$$\forall(\theta, \theta') : p_\theta(x) = p_{\theta'}(x) \implies p_\theta(x, z) = p_{\theta'}(x, z).$$

This means that if we find a parameter vector θ for which $p_\theta(x) = p^*(x)$, we also have that $p_\theta(x, z) = p^*(x, z)$. This leads to the fact that we have found the correct source density distribution $p_\theta(z) = p^*(z)$ and correct conditional distributions $p_\theta(x|z) = p^*(x|z)$ and $p_\theta(z|x) = p^*(z|x)$. The whole VAE model is illustrated in Fig. 1.

In the nonlinear BSS framework, the identifiability has been recently achieved by assuming that the latent sources z have a conditional distribution $p(z|u)$, where $u \in \mathbb{R}^m$ is an auxiliary variable. The auxiliary variable can for example be previous observations in time (Hyvärinen & Morioka, 2017) or current time index (Hälvä & Hyvärinen, 2020; Hyvärinen & Morioka, 2016; Hyvärinen et al., 2019). Similarly, by assuming that the true latent generating model has the form

$$p^*(x, z|u) = p^*(x|z)p^*(z|u), \quad (4)$$

the identifiability can be achieved in the VAE framework, yielding identifiable VAE (iVAE) (Khemakhem et al., 2020). In iVAE, the distribution $p^*(x|z)$ is defined as

$$p^*(x|z) = p_\epsilon^*(x - f(z)),$$

which means that x can be decomposed into $x = f(z) + \epsilon$, where ϵ is an independent noise vector with density p_ϵ . Assuming the non-noisy nonlinear BSS model (3), the distribution p_ϵ can be modelled with Gaussian distribution with infinitesimal variance. The function $f : \mathbb{R}^P \rightarrow \mathbb{R}^S$ is an injective, but possibly nonlinear function. The conditional distribution of latent sources z is assumed to be a part of the exponential family, that is,

$$p_{T, \lambda}(z|u) = \prod_{i=1}^P \frac{Q_i(z_i)}{Z_i(u)} \exp \left[\sum_{j=1}^k T_{i,j}(z_i) \lambda_{i,j}(u) \right], \quad (5)$$

where $Q_i(z_i)$ is a base measure, $Z_i(u)$ is a normalizing constant, $T_i(z_i) = (T_{i,1}(z_i), \dots, T_{i,k}(z_i))^T$ contains sufficient statistics, and $\lambda_i(u) = (\lambda_{i,1}(u), \dots, \lambda_{i,k}(u))^T$ contains the parameters depending on u . The dimension k of each sufficient statistic $T_i(z_i)$ and $\lambda_i(u)$ is assumed to be fixed. The latent components z are identifiable up to permutation and signed scaling under some generally mild conditions on the mixing function f , the sufficient statistics T and the auxiliary variable u . In this study, we construct iVAE assuming Gaussian latent components. Then, for

identifiability, the variances of the latent components are required to vary enough based on the auxiliary variable \mathbf{u} and the mixing function f is required to have continuous partial derivatives. The exact identifiability conditions can be found in [Khemakhem et al. \(2020\)](#).

The iVAE model is similar to the regular VAE model with the exception that iVAE has an additional auxiliary function $\mathbf{w}(\mathbf{u})$ and its parameters θ_w to be estimated, and the encoder $g(\mathbf{x}, \mathbf{u})$ takes both, observations \mathbf{x} and the auxiliary variables \mathbf{u} as an input. The auxiliary function \mathbf{w} maps \mathbf{u} into $\mu_{z|\mathbf{u}}$ and $\sigma_{z|\mathbf{u}}$, which are used to calculate the loss. For iVAE model, ELBO is obtained as

$$\mathcal{L}(\theta|\mathbf{x}, \mathbf{u}) \geq E_{q_{\theta}(z|\mathbf{x}, \mathbf{u})}(\log p_{\theta_h}(\mathbf{x}|\mathbf{z}) + \log p_{\theta_w}(\mathbf{z}|\mathbf{u}) - \log q_{\theta_g}(\mathbf{z}|\mathbf{x}, \mathbf{u})),$$

where $\log p_{\theta_h}(\mathbf{x}|\mathbf{z})$ controls the reconstruction accuracy and $\log p_{\theta_w}(\mathbf{z}|\mathbf{u}) - \log q_{\theta_g}(\mathbf{z}|\mathbf{x}, \mathbf{u})$ is a Kullback–Leibler (KL) divergence between $p_{\theta_w}(\mathbf{z}|\mathbf{u})$ and q_{θ_g} keeping the distributions p_{θ_w} and q_{θ_g} as similar as possible. ELBO is maximized to obtain the estimated parameters $\theta = (\theta_g^T, \theta_h^T, \theta_w^T)^T$. The distributions p_{θ_h} , p_{θ_w} and q_{θ_g} are typically Gaussian distributions, where the functions h , w and g give the mean and variance parameters of the distributions. The distributions can also be other than Gaussian as long as the resampling can be done using the reparametrization trick to allow the backpropagation go through the resampling node. Then, the functions h , w and g do not give mean and variance, but the parameters according the chosen distributions. As we assume Gaussian latent data in this paper, we have $p_{\theta_w} = N(\mathbf{z}|\mu_{z|\mathbf{u}}, \text{diag}(\sigma_{z|\mathbf{u}}))$, $q_{\theta_g} = N(\mathbf{z}|\mu_{z|\mathbf{x}, \mathbf{u}}, \text{diag}(\sigma_{z|\mathbf{x}, \mathbf{u}}))$ and $p_{\theta_h} = N(\mathbf{x}|\mathbf{x}', \beta I)$, where $\beta > 0$ is a small constant as $p_{\theta_h}(\mathbf{x}|\mathbf{z})$ estimates the true distribution $p^*(\mathbf{x}|\mathbf{z})$ with infinitesimal variance. By increasing β , the weight of the reconstruction accuracy in the ELBO decreases. Based on our empirical investigations, we use $\beta = 0.02$ which provides a good balance between the reconstruction error and KL divergence in ELBO, and leads to good performance. [Fig. 1](#) has representations of both VAE and iVAE models and illustrates the difference between the models.

3. iVAE for STBSS

To perform nonlinear spatio-temporal blind source separation using iVAE, the auxiliary variables \mathbf{u} must be selected appropriately. The main assumption for identifiability in spatio-temporal setting is that the variances of the latent components are varying in space and/or in time. This assumption is met by assuming that the latent components are second-order nonhomogeneous, meaning that the second moment of the marginal distribution $p(z_i)$ is not invariant with respect to the location shift in space and/or in time. In addition, the latent components are allowed to be first-order nonhomogeneous, meaning that the components can have nonconstant spatio-temporal trend. The auxiliary variables are constructed in a way that the auxiliary function \mathbf{w} is capable to learn and estimate the mean and the variance vectors of the location of the corresponding multivariate observation. We propose three spatio-temporal iVAE methods; a naive coordinate based method, a segmentation based method extended from spatial iVAE ([Sipilä et al., 2024](#)) and a radial basis function based method utilizing ideas of [Nag, Sun, and Reich \(2023\)](#). Each of the three methods construct the auxiliary data differently based on the spatio-temporal location of the observation. Notice that although in many options below the auxiliary variables are constructed separately for spatial coordinates and temporal coordinates, the auxiliary functions can still learn complex spatio-temporal interactions in the mean and in the variance as the auxiliary functions are modelled by deep neural networks. Furthermore, even though the methods for constructing the auxiliary data are defined here for spatial dimension $D = 2$, the same ideas apply also for higher D .

The approaches presented here are well scalable in terms of computation time. The computation time grows sublinearly with respect to the sample size n (as fewer training epochs are typically needed with larger datasets) and linearly with respect to the dimensions of observed

data, latent data and auxiliary data. However, if the dimension of the auxiliary data and sample sizes are large, the memory usage may grow unless the auxiliary data are formed batch-wise. An in-depth analysis of computational complexity is provided in [Appendix A](#).

3.1. Coordinate based algorithm

In coordinate based iVAE, the preprocessed coordinates are used directly as auxiliary variable. The preprocessed coordinates are obtained by applying min–max normalization to each dimension. The preprocessed coordinates are then

$$\bar{s}_1 = \frac{s_1 - s_1^{\min}}{s_1^{\max} - s_1^{\min}}, \bar{s}_2 = \frac{s_2 - s_2^{\min}}{s_2^{\max} - s_2^{\min}} \text{ and } \bar{t} = \frac{t - t^{\min}}{t^{\max} - t^{\min}},$$

where s_1^{\min} , s_2^{\min} and t^{\min} are the minimum coordinates of the locations of the observations, and s_1^{\max} , s_2^{\max} and t^{\max} are the maximum coordinates of the locations of the observations. The algorithm with the preprocessed coordinates, $\mathbf{u}(s, t) = (\bar{s}_1, \bar{s}_2, \bar{t})^T$, as auxiliary variable is denoted by iVAEC.

3.2. Segmentation based algorithm

In segmentation based iVAE, a spatio-temporal segmentation is used as an auxiliary variable. The spatio-temporal segmentation means that the domain $S \times T$ is divided into m nonintersecting segments $\mathcal{K}_i \in S \times T$ so that $\mathcal{K}_i \cap \mathcal{K}_j = \emptyset$ for all $i \neq j$, $i, j = 1, \dots, m$, and $\cup_{i=1}^m \mathcal{K}_i = S \times T$. By using an indicator function $\mathbb{1}$, the auxiliary variable for the observation $\mathbf{x}(s, t)$ can be written as $\mathbf{u}(s, t) = (\mathbb{1}((s, t) \in \mathcal{K}_1), \dots, \mathbb{1}((s, t) \in \mathcal{K}_m))^T$, where $\mathbb{1}((s, t) \in \mathcal{K}_i) = 1$, if the location (s, t) is within the segment \mathcal{K}_i , and otherwise $\mathbb{1}((s, t) \in \mathcal{K}_i) = 0$. This results into m -dimensional standard basis vector, where the value 1 gives the spatio-temporal segment in which the location of the observation belongs.

If the spatio-temporal domain is large and small segments are used, the dimension m of the auxiliary variable becomes very large. To lower the dimension, the spatial and temporal segmentations can be considered separately. This means that the auxiliary data is composed of m_S spatial segments $S_i \in S$ and m_T temporal segments $T_i \in T$ so that $S_i \cap S_j = \emptyset$ for all $i \neq j$, $i, j = 1, \dots, m_S$, $\cup_{i=1}^{m_S} S_i = S$, $T_i \cap T_j = \emptyset$ for all $i \neq j$, $i, j = 1, \dots, m_T$, and $\cup_{i=1}^{m_T} T_i = T$. Then, the auxiliary variable for the observation $\mathbf{x}(s, t)$ is $\mathbf{u}(s, t) = (\mathbb{1}(s \in S_1), \dots, \mathbb{1}(s \in S_{m_S}), \mathbb{1}(t \in T_1), \dots, \mathbb{1}(t \in T_{m_T}))^T$. The auxiliary variable is $(m_S + m_T)$ -dimensional and has two nonzero entries for each observation. The dimension can be reduced even further by considering also the x -axis and y -axis of the spatial domain separately. Segmentation based auxiliary variables are illustrated in [Fig. 2](#), in which spatial and temporal segmentations are considered separately. We denote the algorithm with all dimensions segmented separately as iVAEs1, with space and time segmented separately as iVAEs2, and with spatio-temporal segmentation as iVAEs3, respectively.

3.3. Radial basis function based algorithm

In radial basis function based iVAE, the auxiliary variable is defined using radial basis functions (see e.g. [Hastie, Tibshirani, Friedman, and Friedman \(2009\)](#)). The idea is that with large number of appropriate radial basis functions, the model incorporates much more spatio-temporal information than by using the coordinates only. Similar ideas have been used recently in [Chen, Li, Reich, and Sun \(2024\)](#), [Nag et al. \(2023\)](#) to perform deep learning based spatial and spatio-temporal predicting by using the spatial and spatio-temporal locations transformed into radial basis functions as input for deep neural networks. Following [Nag et al. \(2023\)](#), we transform spatial and temporal locations separately into radial basis functions. Let $\{o_i^S\}$, $i = 1, \dots, K_S$, where $o_i^S \in S$, be a set of spatial node points, and let $\{o_i^T\}$, $i = 1, \dots, K_T$, where $o_i^T \in T$, be a

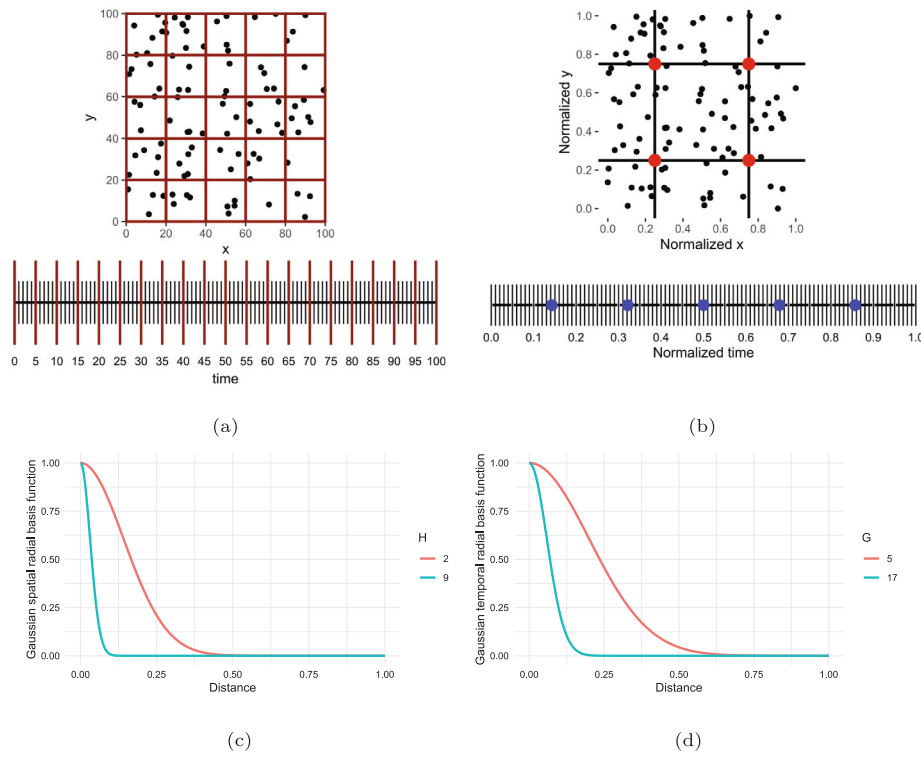


Fig. 2. Illustrations of auxiliary variables of segmentation based iVAE (iVAEs2) (a) and radial basis function based iVAE (b). The top figure of (a) illustrates spatial segmentation, where each segment has size 20×20 producing 25 spatial segments, and the bottom figure illustrates temporal segmentation, where each segment has 5 time points, producing 20 temporal segments. In (b), the black lines in the top figure are the normalized x and y values at $1/(H+2) = 1/4$ and $1/(H+2) + 1/H = 3/4$ formed by resolution level $H = 2$, and the red points represent the produced spatial node points. The blue points represent temporal node points for resolution level $G = 5$. Spatial and temporal Gaussian radial basis functions are illustrated in (c) and (d), respectively. The radial basis functions are functions of distance between node point and a spatial or temporal location.

set of temporal node points. The parameter ζ is a scale parameter. The spatial and temporal radial basis functions are given as

$$v^S(s; \zeta, o_i^S) = v(\|s - o_i^S\|/\zeta) \quad \text{and} \quad v^T(t; \zeta, o_i^T) = v(|t - o_i^T|/\zeta),$$

where v is a kernel function such as the Gaussian kernel $v_G(d) = e^{-d^2}$, or one of the Wendland kernels (Wendland, 1995) such as

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

Following Nychka, Bandyopadhyay, Hammerling, Lindgren, and Sain (2015), we use a multi-resolution approach to form the spatial and temporal radial basis functions. Each resolution level is composed of its own number of evenly spaced node points and own scaling parameter. A low level resolution with small number of node points and large value of the scaling parameter aims to capture large-scale spatial or temporal dependencies, while a high level resolution with many node points and small scaling parameter aims to find finer details of the dependence structure.

To form the radial basis functions, we first preprocess the spatial and temporal locations to range $[0, 1]$ using min-max normalization. A H -level spatial resolution is formed of evenly spaced grid of node points $\{o_i^S\}$ with spacing $1/H$ and an offset $1/(H+2)$ before the first node point, meaning that H -level resolution has the node points $\{(i, j) : i, j \in \{\frac{1}{H+2}, \frac{1}{H+2} + \frac{1}{H}, \dots, 1 - \frac{1}{H+2}\}\}$. For example 2-level spatial resolution is then composed of the node points $\{(i, j) : i, j \in \{0.25, 0.75\}\}$. Similarly, a G -level temporal resolution is formed of evenly spaced one dimensional node points $\{o_i^T\}$ with spacing $1/G$ and an offset $1/(G+2)$, meaning that G -level temporal resolution is composed of the node points $\{\frac{1}{G+2}, \frac{1}{G+2} + \frac{1}{G}, \dots, 1 - \frac{1}{G+2}\}$. As the scaling parameters ζ_H and ζ_G , for spatial and temporal radial basis functions, we use $\zeta_H = \frac{1}{2.5H}$ following Nychka et al. (2015) and $\zeta_G = \frac{|o_1^T - o_2^T|}{\sqrt{2}}$ following Nag et al. (2023). Spatial and temporal node points and radial basis functions are

illustrated in Fig. 2 for $H = 2$ spatial resolution, producing 4 spatial radial basis functions, and $G = 5$, producing 5 temporal radial basis functions. In practice, multiple spatial and temporal resolution levels, such as $H = (H_1, H_2) = (2, 9)$, and $G = (G_1, G_2, G_3) = (9, 17, 37)$, should be used to capture both large scale and finer dependencies. An advantage of using radial basis functions as auxiliary variables instead of spatio-temporal segments is that by using radial basis functions, iVAE's auxiliary function provides a smooth spatio-temporal trend and variance functions, which can be used later for further analysis such as for prediction purposes. The radial basis function based iVAE is denoted as iVAE_r in the rest of the paper.

4. Simulation studies

The aim of this section is to demonstrate and compare the performances of iVAE methods using simulation studies and to discover how various types of nonstationarity in variance affect the performance. The section begins with a short review of some common procedures for generating spatio-temporal data and ways to introduce nonstationarity in it. The remainder of the section contains a large simulation study showing the unmixing performances of the iVAE methods under different types of nonstationarity scenarios, and then introduces two methods to estimate the number of latent signals. Finally, the latent dimension estimation methods are illustrated in a small simulation study. All simulations can be reproduced using R 4.3.0 (R Core Team, 2023) together with R packages fastICA (Marchini, Heaton, & Ripley, 2021), SpaceTimeBSS (Muehlmann, Piccolotto, Cappello, De Iaco, & Nordhausen, 2022) and NonlinearBSS. NonlinearBSS package contains R implementations of all proposed spatio-temporal iVAE variants, and is available in <https://github.com/mikasip/NonlinearBSS>. The simulations were executed on the CSC Puhti cluster, a high-performance computing environment.

4.1. Nonstationary spatio-temporal data generation

Spatio-temporal data are typically composed of n_s spatial locations and n_t temporal points, making the total number of observations $n = n_s n_t$ usually very high. The observations are often collected regularly, for example daily or hourly, by some monitoring stations in different locations. This makes the observed data quickly very dense in time but more sparse in space. To study the properties of the models under the nature of real life spatio-temporal data, generating large datasets with various spatio-temporal covariance models is required. In the following simulations, we exploit a computationally efficient vector autoregressive process, see for example (Papalexiou & Serinaldi, 2020; Sigrist, Künsch, & Stahel, 2012; Xu & Gardoni, 2018; Yan, Huang, & Genton, 2021), and a simplified version of improved latent space approach (ILSA) (Xu & Gardoni, 2018) to generate nonstationary spatio-temporal data.

Assume spatial field at time $t = 1, \dots, n_t$ to be $\delta(t) = (\delta(s_1, t), \dots, \delta(s_{n_s}, t))^T$, where s_1, \dots, s_{n_s} are the spatial locations in the spatio-temporal field. The vector autoregressive process can be written as

$$\delta(t) = \sum_{r=1}^R \rho_r \mathbf{K}_r \delta(t-r) + \epsilon_\delta(t), \quad (6)$$

where $r = 1, \dots, R$ is an autoregressive order, ρ_r is r th baseline autoregressive coefficient, \mathbf{K}_r is a $n_s \times n_s$ spatial kernel matrix determining the change of temporal correlation with spatial locations, and $\epsilon_\delta(t)$ is a n_s -dimensional noise vector with covariance $C(\epsilon_\delta(s, t), \epsilon_\delta(s', t))$ with $s, s' \in \{s_1, \dots, s_{n_s}\}$.

With a simplified version of ILSA, one can generate nonstationary spatio-temporal data by using vector autoregressive process (6) as formulated next. Let $\tilde{s}(s) = [\tilde{s}_1, \dots, \tilde{s}_d]$ be a d -dimensional transformation of the original coordinate s , where the transformed coordinates \tilde{s}_i , $i = 1, \dots, d$, are called regressors or latent coordinates. Let $d_{s_i s_j} = [\|s_1^1 - s_1^2\|, \|s_2^1 - s_2^2\|]^T$, $d_{\tilde{s}_i \tilde{s}_j} = [\|\tilde{s}_1^1 - \tilde{s}_1^2\|, \dots, \|\tilde{s}_d^1 - \tilde{s}_d^2\|]^T$ and V to be any stationary covariance function. Simplified ILSA has the formulations

$$\mathbf{K}_{r\{i,j\}} = \left| \begin{matrix} \theta_{s,r} & \\ & \theta_{\tilde{s},r} \end{matrix} \right|^{-\frac{1}{2}} \exp \left(- \left[\begin{matrix} d_{s_i s_j} & d_{\tilde{s}_i \tilde{s}_j} \end{matrix} \right] \begin{bmatrix} \theta_{s,r} \\ \theta_{\tilde{s},r} \end{bmatrix} \begin{bmatrix} d_{s_i s_j} \\ d_{\tilde{s}_i \tilde{s}_j} \end{bmatrix} \right),$$

$$C(\epsilon_\delta(s, t), \epsilon_\delta(s', t)) = \sigma[\tilde{s}(s), s, t] \sigma[\tilde{s}(s'), s', t] V(Q_t) \quad (7)$$

$$\text{where } Q_t = \left(\begin{bmatrix} d_{s_i s_j} & d_{\tilde{s}_i \tilde{s}_j} \\ \theta_s & \theta_{\tilde{s}} \end{bmatrix} \begin{bmatrix} d_{s_i s_j} \\ d_{\tilde{s}_i \tilde{s}_j} \end{bmatrix} \right)^{\frac{1}{2}}$$

and $\theta_{s,r} = \text{diag}(\theta_{s_1,r}, \theta_{s_2,r})$, $\theta_{\tilde{s},r} = \text{diag}(\theta_{\tilde{s}_1,r}, \dots, \theta_{\tilde{s}_d,r})$, $\theta_s = \text{diag}(\theta_{s_1}, \theta_{s_2})$ and $\theta_{\tilde{s}} = \text{diag}(\theta_{\tilde{s}_1,r}, \dots, \theta_{\tilde{s}_d,r})$ are diagonal matrices giving scaling parameters for the spatial coordinates and for the latent coordinates. The function Q_t transforms the original coordinates based on the scaling parameters and the latent coordinates. By using this approach, one can easily introduce complex, nonstationary and nonseparable covariance structures through latent coordinates \tilde{s} , time varying spatial kernel matrices \mathbf{K}_r and nonstationary variance function σ . In the following simulations, we are mainly interested in having nonstationarity in the variance as that is required for the identifiability of the latent components.

4.2. Finite sample efficiencies

In this section, four different iVAE configurations – regular VAE, symmetric FastICA (FICA) (Hyvärinen, 1999) with hyperbolic tangent nonlinearity, and STBSS – are compared using simulated spatio-temporal data. Although FICA is not designed for spatio-temporal data or nonlinear mixing, it is included as a linear baseline for data with nonstationary variances. STBSS, developed for stationary spatio-temporal data and linear mixing, serves as a spatio-temporal baseline.

While there are several deep learning-based approaches for nonlinear BSS in the literature, see Ansari et al. (2023) for a recent review, most lack identifiability and focus on acoustic data, which primarily exhibits serial dependence, making them suboptimal for spatio-temporal

data. Nonetheless, we include regular VAE as an unidentifiable deep learning baseline.

The aim of the simulations is to identify how the proposed iVAE methods perform as compared to other existing methods under various types of nonstationary spatio-temporal data, and how the type of nonstationary affects the performance. To identify how the reduction of either temporal or spatial observations affect the performance of the algorithms, we consider three sample sizes composed of n_s spatial locations and n_t temporal observations for each spatial location. The sample dimensions considered are $(n_s, n_t) = (150, 300)$, $(n_s, n_t) = (50, 300)$ and $(n_s, n_t) = (150, 75)$ yielding $n = 45000$, $n = 15000$ and $n = 11250$ observations, respectively. We generate the latent data z according to six different simulation settings. In some settings the nonstationarity is introduced only in time, in some settings only in space, and in some settings both in space and in time. In each simulation setting, n_s spatial locations s_i are sampled uniformly in spatial domain $S = [0, 1] \times [0, 1]$, and for each spatial location s_i , n_t observations $x(s_i, t)$ are generated. The true latent dimension is $P = 5$ and the dimension of the observations is $S = 8$. Every setting is repeated 500 times for each sample size and for each algorithm. Finally, each trial is repeated using three increasingly nonlinear mixing functions as described hereafter. The first three simulation settings are more simple ones, followed by three more complex ones which utilize the ILSA framework. The simulation settings and the mixing procedure are defined in the following. After introducing the data generation of the settings, the motivation behind each setting is carefully explained.

Setting 1. The latent spatio-temporal field consists of three clusters in space and five segments in time yielding 15 spatio-temporal clusters, each of which has their own unique diagonal covariance matrix and unique mean vector. For k th cluster, $k = 1, \dots, 15$, the covariance matrix is given as $\mathbf{C}_k = \text{diag}(\sigma_{1,k}, \dots, \sigma_{5,k})$, where $\sigma_{i,k} \sim \text{Unif}(0.1, 5)$ and unique mean vector is given as $\boldsymbol{\mu}_k = (\mu_{1,k}, \dots, \mu_{5,k})^T$, where $\mu_{i,k} \sim \text{Unif}(-5, 5)$, $i = 1, \dots, 5$.

Setting 2. The latent spatio-temporal field consists of 10 segments in time. The latent components are simulated by generating first Gaussian spatial data with Matern covariance function using unique parameters (ν_i, ϕ_i) for each component $i = 1, \dots, 5$, and then adding Gaussian iid data with unique covariance matrix and mean vector for each time segment. The Matern parameters are $(\nu_1, \phi_1) = (0.5, 0.30)$, $(\nu_2, \phi_2) = (0.1, 0.25)$, $(\nu_3, \phi_3) = (1, 0.35)$, $(\nu_4, \phi_4) = (2, 0.20)$, $(\nu_5, \phi_5) = (0.25, 0.15)$ and the parameters for the time segment $k = 1, \dots, 10$ are $\boldsymbol{\mu}_k = (\mu_{1,k}, \dots, \mu_{5,k})^T$, where $\mu_{i,k} \sim \text{Unif}(-0.3, 0.3)$, and $\boldsymbol{\Sigma}_k = \text{diag}(\sigma_{1,k}, \dots, \sigma_{5,k})$, where $\sigma_{i,k} \sim \text{Unif}(0, 0.4)$, $i = 1, \dots, 5$.

Setting 3. The latent spatio-temporal field consists of five clusters in space and follows AR1 model. In k th cluster, $k = 1, \dots, 5$, the latent components z_i , $i = 1, \dots, 5$, are generated as

$$z_i(s, t+1) = \rho_{i,k} z_i(s, t) + \epsilon_{i,k,t},$$

$$\epsilon_{i,k,t} \sim N(\mu_{i,k}, \sigma_{i,k}),$$

where $t = 1, \dots, n_t - 1$ and $z_i(s, 1) \sim N(\mu_{i,k}, \sigma_{i,k})$. Each cluster has unique parameters $\boldsymbol{\rho}_k = (\rho_{1,k}, \dots, \rho_{5,k})^T$, $\boldsymbol{\mu}_k = (\mu_{1,k}, \dots, \mu_{5,k})^T$ and $\boldsymbol{\sigma}_k = (\sigma_{1,k}, \dots, \sigma_{5,k})^T$ generated as $\rho_{i,k} \sim \text{Unif}(0.05, 0.95)$, $\mu_{i,k} \sim \text{Unif}(-1, 1)$ and $\sigma_{i,k} \sim \text{Unif}(0.1, 5)$.

Settings 4–6. The latent spatio-temporal field is generated using ILSA framework. Each setting has the same highly nonstationary covariance structure. In addition, Setting 4 has a variance σ changing in space, Setting 5 has a variance changing in time, and Setting 6 has a variance changing both in space and in time. The latent coordinates $\tilde{s} = (\tilde{s}_1, \tilde{s}_2)^T$ are transformed from the spatial coordinates $s = (s_1, s_2)^T$ by using a swirl-like coordinate transformation according to Papalexiou, Serinaldi, and Porcu (2021) given by

$$\tilde{s}_1 = (s_1 - s_1^*) \cos \left(\eta \exp \left(- \left(\frac{h^*}{b_{\text{swirl}}} \right)^2 \right) \right) - (s_2 - s_2^*) \sin \left(\eta \exp \left(- \left(\frac{h^*}{b_{\text{swirl}}} \right)^2 \right) \right) + s_1^*,$$

Table 1
The ILSA parameters and coordinate deformation parameters for Settings 4–6.

| | $\theta_{s,j}$ | $\theta_{s,i}$ | θ_s | θ_s | ρ_1 | s^* | b_{swirl} | η | ν | ϕ |
|-----|----------------|----------------|------------|-------------|----------|------------|--------------------|----------|-------|--------|
| IC1 | (6, 4) | (7, 7) | (0.2, 0.7) | (0.7, 0.2) | 0.9 | (0.5, 0.5) | 0.7 | 1.8π | 0.25 | 0.5 |
| IC2 | (3, 6) | (4, 7) | (0.7, 0.2) | (0.25, 0.5) | 0.8 | (0.7, 0.7) | 0.4 | 1.2π | 0.2 | 0.9 |
| IC3 | (3, 3) | (6, 3) | (0.5, 0.5) | (0.7, 0) | 0.7 | (0.3, 0.3) | 0.2 | 2π | 0.05 | 1.5 |
| IC4 | (7, 3) | (2, 6) | (0.2, 0.4) | (0.3, 0.7) | 0.6 | (0.7, 0.3) | 1 | 0.5π | 0.1 | 0.25 |
| IC5 | (2, 1) | (6, 2) | (0.3, 0.3) | (0, 0.7) | 0.5 | (0.3, 0.7) | 0.9 | 0.9π | 0.15 | 1 |

$$\tilde{s}_2 = (s_1 - s_1^*) \sin\left(\eta \exp\left(-\left(\frac{h^*}{b_{\text{swirl}}}\right)^2\right)\right) - (s_2 - s_2^*) \cos\left(\eta \exp\left(-\left(\frac{h^*}{b_{\text{swirl}}}\right)^2\right)\right) + s_2^*,$$

where $s^* = (s_1^*, s_2^*)$ is the centre point of the deformation, $h^* = \|s - s^*\|$ is Euclidean distance between the original location and the centre point, η is a rotation angle, and b_{swirl} is a scaling parameter controlling the magnitude of the swirl. Each latent component has their own set of deformation parameters. The stationary covariance function V in (7) is the Matern covariance function with parameters (ν_i, ϕ_i) for all Settings 4–6. The deformation parameters, ILSA parameters and Matern parameters mutual for Settings 4–6 are given in Table 1. The autoregressive order is $R = 1$ for all settings.

In Setting 4, we have $\sigma[\tilde{s}(s), s, t] = \exp(\theta_{\sigma_s}^i (\tilde{s}_1 - 0.5))$, where $\theta_{\sigma_s}^i$ is the scaling parameter of variance in space for i th latent component. This means that the variance of the latent fields vary in space based on the first latent coordinate. The variance scaling parameters for the latent components z_i , $i = 1, \dots, 5$, are $\theta_{\sigma_s}^1 = 1$, $\theta_{\sigma_s}^2 = 2$, $\theta_{\sigma_s}^3 = 3$, $\theta_{\sigma_s}^4 = -1$ and $\theta_{\sigma_s}^5 = -2$.

In Setting 5, the variances of the latent fields are changing in time. We set $\sigma[\tilde{s}(s), s, t] = \exp(\sin((t + \theta_{\sigma_{t1}}^i) + \theta_{\sigma_{t2}}^i)/2)$, where $\theta_{\sigma_{t1}}^i$ and $\theta_{\sigma_{t2}}^i$ are variance coefficient and variance scaling parameter in time for i th latent component. The parameters $(\theta_{\sigma_{t1}}^i, \theta_{\sigma_{t2}}^i)$ for the latent components z_i , $i = 1, \dots, 5$, are $(\theta_{\sigma_{t1}}^1, \theta_{\sigma_{t2}}^1) = (50, 0.1)$, $(\theta_{\sigma_{t1}}^2, \theta_{\sigma_{t2}}^2) = (0, 0.05)$, $(\theta_{\sigma_{t1}}^3, \theta_{\sigma_{t2}}^3) = (100, 0.005)$, $(\theta_{\sigma_{t1}}^4, \theta_{\sigma_{t2}}^4) = (20, 0.01)$ and $(\theta_{\sigma_{t1}}^5, \theta_{\sigma_{t2}}^5) = (10, 0.03)$.

In Setting 6, the variances of the latent fields are changing in space and in time. We set $\sigma[\tilde{s}(s), s, t] = \exp(\theta_{\sigma_s}^i (\tilde{s}_1 - 0.5) + \sin((t + \theta_{\sigma_{t1}}^i) + \theta_{\sigma_{t2}}^i)/2)$. The parameters $\theta_{\sigma_s}^i, \theta_{\sigma_{t1}}^i, \theta_{\sigma_{t2}}^i$ for the latent fields z_i , $i = 1, \dots, 5$, are identical as in Settings 4 and Setting 5.

Setting 1 has the simplest latent fields by having a diagonal spatio-temporal covariance for each latent field. The variance and mean are changing explicitly between the spatio-temporal clusters as is assumed for segmentation based iVAE. This setting is a spatio-temporal variant of the simulation setting used in time series context in Hyvärinen and Morioka (2016), Khemakhem et al. (2020), where the latent components had multiple temporal segments with unique mean and/or variance parameters. Settings 2 and 3 are still relatively simple with no spatio-temporal interaction in the latent fields. Setting 2 is used to compare performances in cases where latent fields are stationary in space, but variance is changing over time. Setting 3 illustrates a scenario where the latent fields are stationary in time, but the variance is changing over the clusters in space. By having less variability in the variance, Settings 2 and 3 should be less optimal for iVAE. Settings 4–6 have latent fields with a complex spatio-temporal covariance model and strong spatio-temporal interaction. The variance is not changing explicitly over segments, but instead through a nonstationary covariance function. In Setting 4, the latent fields have smoothly changing nonstationary variance in space, but the variance is stationary in time, and in Setting 5, the variance is nonstationary in time, but stationary in space. Setting 6 introduces nonstationarity in variance both in space and in time. With Settings 4–6 the aim is thus to find out how the presence of nonstationarity in variance affects the performances of iVAE methods in settings with more realistic and more complex spatio-temporal structures.

Nonlinear mixing functions. The mixing function f_L is generated using multilayer perceptron (MLP) following Hyvärinen and Morioka (2016), Hyvärinen et al. (2019), Khemakhem et al. (2020). Here L denotes the number of mixing layers used in MLP. To obtain an injective and differentiable mixing function, each layer of MLP has $S = 8$ hidden units with the activation function ω_i being either linear or exponential linear unit (ELU). The matrices \mathbf{B}_i , $i = 1, \dots, L$, in the mixing procedure are normalized to have unit length row and column vectors to guarantee that none of the independent components vanish in the mixing process. The mixing function f_L is defined as

$$f_L(\mathbf{z}) = \begin{cases} \omega_L(\mathbf{B}_L \mathbf{z}), & L = 1, \\ \omega_L(\mathbf{B}_L f_{L-1}(\mathbf{z})), & L \in \{2, 3, \dots\}, \end{cases}$$

where \mathbf{B}_1 is a 8×5 matrix and the other matrices, \mathbf{B}_i , $i \neq 1$, are 8×8 matrices. In simulations we use linear activation $\omega_L(x) = x$ for the last layer and ELU activation

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

$i = 1, \dots, L - 1$, for the other layers. By this procedure, with the number of mixing layers $L = 1$, we obtain $S = 8$ linear mixtures of the independent components. When the number of mixing layers increase, the mixtures become increasingly nonlinear. In simulations, we consider three different mixing functions with the number of mixing layers $L = 1, 3, 5$.

Model specifications. All iVAE models are set up with encoder, decoder and auxiliary function with three hidden layers in each. The hidden layers consist of 128 neurons and leaky rectified linear unit (leaky ReLU) activation (Maas, Hannun, Ng, et al., 2013). iVAEs1, iVAEs2 and iVAEs3 use 4×4 spatial segmentation, resulting a grid of $m_S = 16$ equally sized squares. The temporal segmentation is done by dividing the temporal domain to equally sized segments of length 5. This results the number of temporal segments $m_T = 60$ when $n_t = 300$ and $m_T = 15$ when $n_t = 75$. For iVAER, we use spatial resolution levels $H = (2, 9)$, and temporal resolution levels $G = (9, 17, 37)$. The iVAE models are trained for 60 epochs when $(n_s, n_t) = (150, 300)$, for 120 epochs when $(n_s, n_t) = (50, 300)$ and for 150 epochs when $(n_s, n_t) = (150, 75)$. The number of epochs is increased when the sample size is decreased, as the number of training steps in each epoch is lower for the smaller sample size. For all sample sizes, the number of epochs are selected large enough to guarantee that the training converges. All iVAE models use learning rate of 0.001 with polynomial decay of second-order over 10000 training steps, where the learning rate after the first 10000 training steps is 0.0001. VAE uses similar parameters as iVAE, but it does not use any auxiliary data or have an auxiliary function. The STBSS model is fitted with multiple different kernel settings, and the best one is selected, which is having two spatial ring kernels (0, 0.15) and (0.15, 0.3) and time lag of 1. For more about STBSS and its kernel settings, see Muehlmann et al. (2023).

Performance index. To measure the performance of the methods, the mean correlation coefficient (MCC) is used following the previous studies, e.g., Hälvä and Hyvärinen (2020), Hyvärinen and Morioka (2017), Hyvärinen et al. (2019), Sipilä et al. (2024). MCC is a function of the correlation matrix $\mathbf{\Omega} = \text{Cor}(\mathbf{z}, \hat{\mathbf{z}})$ between the true latent components \mathbf{z} and the estimated ones $\hat{\mathbf{z}}$. MCC is calculated as

$$\text{MCC}(\mathbf{\Omega}) = \frac{1}{P} \sup_{\mathbf{P} \in \mathcal{P}} \text{tr}(\mathbf{P} \text{abs}(\mathbf{\Omega})), \quad (8)$$

where \mathcal{P} is a set of all possible $P \times P$ permutation matrices, $\text{tr}(\cdot)$ is the trace of a matrix and $\text{abs}(\cdot)$ denotes taking elementwise absolute values of a matrix. MCC gets values in range $[0, 1]$, where the optimal value 1 is obtained when the estimated sources are correlated perfectly up to their signs with the true sources.

Results. The simulation results are provided in Fig. 3 for $(n_s, n_t) = (150, 300)$ and in Figs. B.8 and B.9 in the Appendix B for $(n_s, n_t) =$

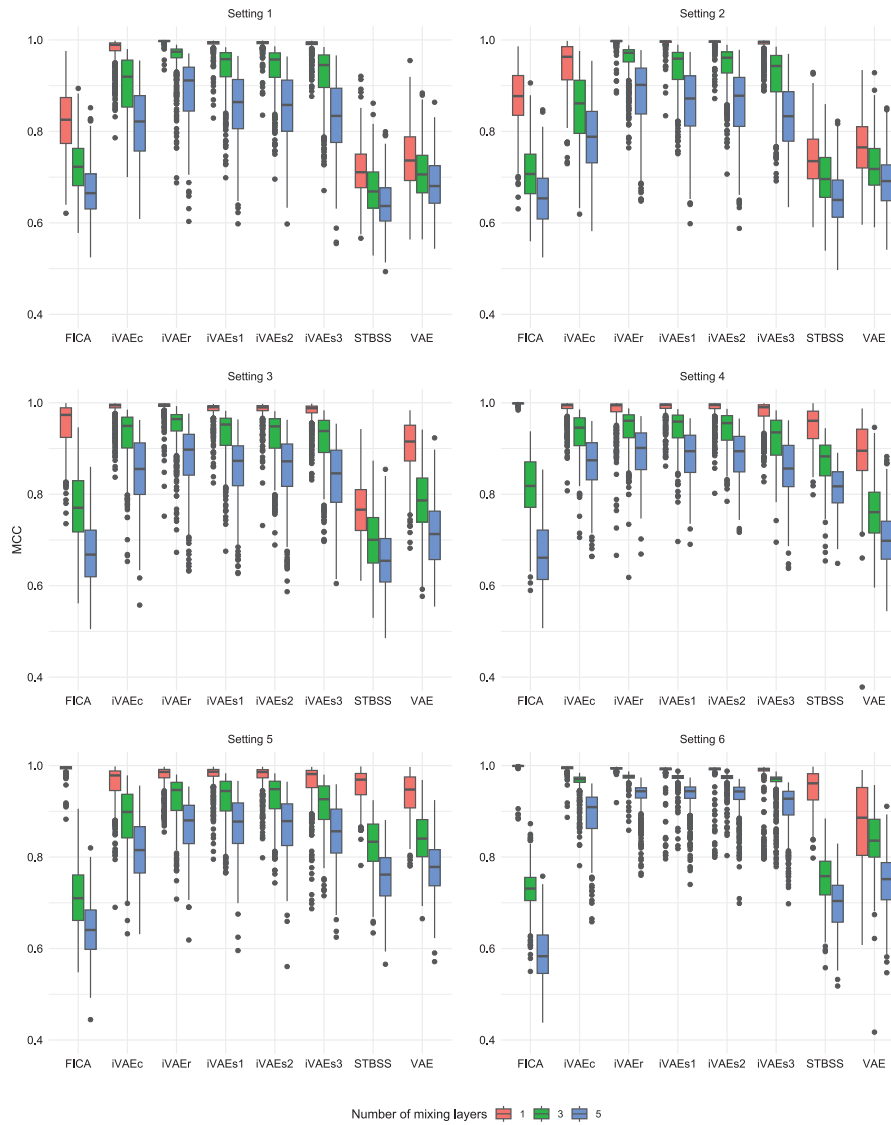


Fig. 3. Mean correlation coefficients of 500 trials for Settings 1–6 for sample size with the number of spatial locations $n_s = 150$ and the number of temporal observations $n_t = 300$.

(50, 300) and $(n_s, n_t) = (150, 75)$, respectively. Based on the results, it is clear that only the iVAE methods are capable of recovering sources through nonlinear unmixing environment. The performances of iVAE methods are better, when the sample size grows, although the differences are small in some settings. The performance of iVAEc is slightly worse than the performances of the other iVAE methods in every setting. FICA performs well in the linear mixing settings, when the latent fields do not contain trend in mean. In nonlinear settings, its performance drops dramatically in all simulation settings. Similar behaviour is present for STBSS, although the performance in zero mean settings does not reach FICA. This is not surprising as STBSS is developed for stationary spatio-temporal random fields. VAE performs poorly in almost all settings, which is expected as the model is not identifiable.

In Settings 1–3 all iVAE methods outperform FICA, STBSS and VAE. In these settings, the best performing method is iVAEr, followed by iVAEs1, iVAEs2 and iVAEs3, respectively. They all perform very well under the linear mixing, but when the number of mixing layers is increased, iVAEr outperforms the three other methods. iVAEs1 and iVAEs2 have very similar performance, and they perform slightly better than iVAEs3. The performance of iVAEc is worse than performances of other iVAE methods, especially in Setting 2. FICA performs relatively well in Setting 3 under the linear mixing, but the performance is poor

in other settings. STBSS fails to recover the latent fields in Settings 1–3 even under the linear setting. VAE fails in Settings 1 and 2, but performs moderately in Setting 3 under linear mixing.

In Settings 4–6, the best method under linear mixing is FICA, but its performance drops considerably in nonlinear settings. In case of nonlinear mixing, the best methods are iVAEs1 and iVAEs2 and iVAEr followed by iVAEs3 and iVAEc, in the order from best to worst. iVAEs1, iVAEs2 and iVAEr perform nearly as well as FICA in linear setting and keep up their good performance also in nonlinear settings. STBSS performs rather well under linear mixing, but is still worse than FICA and iVAE variants. VAE has slightly lower performance than STBSS under linear mixing, but it also fails under nonlinear mixing. In Setting 5, when $n_t = 75$, the performances of all the methods drop considerably. This is probably due to the fact that in Setting 5, the variance is varying less over the whole temporal domain when the number of time points is lower. The best performing methods, when $n_t = 75$, are iVAEs1 and iVAEs2 in both linear and nonlinear cases. In Setting 6, the performance of iVAE methods drop only slightly when the number of mixing layers is increased, especially when the sample size is high. The best performing methods for nonlinear mixing environment are iVAEr, iVAEs1 and iVAEs2.

In general, if the variability of the variance remains the same, increasing the sample size improves the results only a little. The differences in MCC between the smallest and the largest sample sizes are between 0.005 and 0.03 for all iVAE methods and all settings except the Setting 5, which has lower MCCs when $n_t = 75$. Increasing the variability of the variance has stronger impact, which is evident when the results of Settings 4, 5 and 6 are compared. The average MCCs are higher and the results are more consistent for Setting 6 than for Settings 4 and 5. Based on the results, the best three models are iVAEr, iVAEs1 and iVAEs2. Compared to segmentation based iVAE, iVAEr has an advantage of estimating a smooth spatio-temporal trend and variance functions, which are provided by the auxiliary function. These can be useful for further analysis or for prediction purposes, as is demonstrated in Section 5. Taking the previous facts into account, we consider the best method to be iVAEr.

Radial basis function parameter sensitivity. To examine the sensitivity of iVAEr to the choice of resolution levels for forming radial basis functions, we conduct an additional small-scale simulation by replicating Setting 6 with $n_s = 150$ and $n_t = 300$, using iVAEr with three different radial basis function parameter configurations. The first configuration uses resolution levels $H = (2)$ and $G = (9)$, resulting in a total of 13 basis functions. The second configuration, used in previous simulations, has $H = (2, 9)$ and $G = (9, 17, 39)$, producing 150 basis functions. The third configuration uses $H = (2, 9, 17)$ and $G = (9, 17, 39, 99)$, producing 538 basis functions.

The results, presented in Fig. B.10 in Appendix B, indicate that using too few radial basis functions leads to lower performance, as seen with the first setting ($H = (2), G = (9)$). The other two settings produced nearly identical results, suggesting that the algorithm is not highly sensitive to the choice of resolution levels, provided the number of spatial and temporal basis functions is sufficient. As a general guideline, we recommend starting with resolution levels $H = (2, 9)$ and $G = (9, 17, 39)$ for good performance with relatively low computational complexity. Additionally, it is advantageous to use multiple varying spatial and temporal resolution levels (e.g., $H = (2, 9)$ instead of $H = (10)$) to capture both large-scale dependencies and finer dependency structures.

4.3. Latent dimension estimation

In the previous section, we demonstrated through simulations that when the number of latent components P is known, iVAE effectively recovers these components. However, in practical applications, the true number of latent components is often unknown and must be estimated. Accurately determining the dimension is critical; too few components may lead to the omission of vital information, flawed interpretations, and erroneous predictions. Conversely, too many components can complicate interpretation and typically result in overly noisy predictions. One of the reasons for the success of BSS is that it generally simplifies interpretability and enhances predictions, which is particularly valuable in handling complex data such as in spatio-temporal contexts.

Note that the estimation of the number of components in BSS models with linear mixing has only been considered recently. These approaches typically handle iid data, or temporal or spatial data, but, to the best of our knowledge, they have not yet been applied to spatio-temporal data. Moreover, these methods often rely on eigenvalues of specially constructed scatter matrices, which precludes their extension to nonlinear scenarios. For more details on these approaches, see Luo and Li (2016, 2021), Muehlmann, Bachoc, Nordhausen, and Yi (2024), Nordhausen, Taskinen, and Virta (2022), Radojčić and Nordhausen (2024), Virta and Nordhausen (2021), Yi and Nordhausen (2023) and the references therein.

To facilitate the estimation of the number of components within our framework, this section proposes two alternatives: a visual procedure to select the latent dimension and a more formal latent dimension estimation method. We illustrate these methods using simulations. In all simulations, we adopt two setups: one where $P = 5$ and $S = 8$, as

Table 2

The proportions of the estimated latent dimensions R for setups where $P = 5, S = 8$ (a) and $P = 10, S = 15$ (b). Each setup was repeated 100 times for each number of mixing layers $L = 1, 3, 5$. The latent dimensions were estimated using the AIC based method.

| (a) | | |
|-----|-------|-------|
| L | R = 4 | R = 5 |
| 1 | 0.00 | 1.00 |
| 3 | 0.00 | 1.00 |
| 5 | 0.03 | 0.97 |

| (b) | | | |
|-----|-------|--------|--------|
| L | R = 9 | R = 10 | R = 11 |
| 1 | 0.00 | 0.99 | 0.01 |
| 3 | 0.00 | 0.99 | 0.01 |
| 5 | 0.07 | 0.92 | 0.01 |

in the previous simulations, and another where $P = 10$ and $S = 15$. In both setups, the latent components are generated as in Setting 6 of the previous section with randomly generated parameters from appropriate uniform distributions, and the observed data is generated using the same mixing process as in the previous simulations. For these simulations, we employ only iVAEr with the same parameters as in the previous section and train it for 60 epochs in each trial.

Visual knee point detection. As iVAE is capable of estimating injective mixing functions, it is possible to fit multiple iVAE models using the latent dimension $R = 2, \dots, S$ and compare ELBOs of fitted models. The likelihood has its maximum when the true latent components are found, and hence, ELBO tends to increase rapidly when $R < P$ and stay approximately the same when $R \geq P$. Because of this behaviour, when ELBOs are plotted against the selected latent dimensions $R = 2, \dots, S$, a “knee” point is visible at the point of the correct latent dimension. The knee point detection method based on ELBO was also discussed in Khemakhem et al. (2020). The knee point behaviour is illustrated in Fig. 4 for the both setups. Here, the number of mixing layers $L = 3$ were used.

AIC based on ELBO. A natural approach for automatic model selection is to compare Akaike information criteria (AIC) between different models. In case of latent dimension selection, we are only interested in the latent dimension R , and treat the other parameters of the model as nuisance. Hence, we compare the profile AICs (pAIC, see e.g. Xu, Vaida, and Harrington (2009)) calculated as

$$\text{pAIC} = -2\log(\mathcal{L}(R|\mathbf{x}, \mathbf{u}; \hat{\theta}_R)) + 2R,$$

where $\mathcal{L}(R|\mathbf{x}, \mathbf{u}; \hat{\theta}_R)$ is the profile likelihood in which all the parameters θ excluding the latent dimension R have been profiled out. The data log likelihood is intractable for the generative model (4) which is why we cannot use pAIC directly. However, if the profile log likelihood $\log(\mathcal{L}(R|\mathbf{x}, \mathbf{u}; \hat{\theta}_R))$ is replaced with profiled ELBO, we obtain an upper bound for pAIC:

$$\text{uAIC} = -2\text{ELBO}(R|\mathbf{x}, \mathbf{u}; \hat{\theta}_R) + 2R,$$

which can be used for model selection. Similarly as for regular AIC, the latent dimension which produces the lowest uAIC value is selected. To demonstrate the method, we simulated 100 datasets from both setups, $(P, S) = (5, 8)$ and $(P, S) = (10, 15)$, and then fitted iVAEr for each possible latent dimension $R = 2, \dots, S$ and selected the latent dimension based on uAIC. The proportions of the selected dimensions are provided in Table 2 for both setups.

In both setups, AIC based method estimated well the latent dimension for each mixing function. When the number of mixing layers is $L = 1$ or $L = 3$, the true latent dimension was obtained nearly every time (100 out of 100 times for setup with $(P, S) = (5, 8)$ and 99 out of 100 times for setup with $(P, S) = (10, 15)$). When $L = 5$, the method underestimated the latent dimension in three trials out of 100. In setup

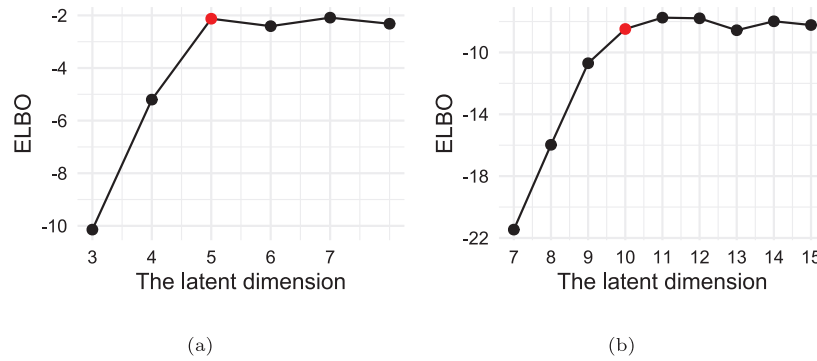


Fig. 4. The lower bounds of the data log likelihoods plotted for different latent dimensions. The true latent dimension is marked with red. The first figure (a) has the setting with $P = 5$ and $S = 8$, and the second one (b) has the setting with $P = 10$ and $S = 15$. In second figure, the latent dimensions $R = 2, \dots, 6$ are cut out to make the differences more visible for larger latent dimensions.

with $(P, S) = (10, 15)$, the method overestimated the latent dimension once for each mixing function. Based on the results, uAIC can thus be seen as a promising metric for automatic latent dimension selection.

5. Real data example

Due to the best overall performance of the proposed radial basis function based spatio-temporal iVAE method, iVAEr, we now demonstrate how to apply it using a meteorological dataset collected from Veneto region in Italy. The data are collected over 23 years (2000–2022) from 101 different meteorological stations. The data consist of weekly data of evapotranspiration level (ET_0, in mm), minimum and maximum temperature (T_min and T_max, respectively, in °C), minimum and maximum humidity (H_min and H_max, respectively, in %), the average wind velocity (m/s) and precipitation (mm). Therefore, we have $S = 7$. We remove 11 stations and all time points before week 28 of year 2005 to have as little missing values as possible. After this, we have data on 90 stations and 892 time points. The remaining 133 rows with missing values are imputed using CUTOFF method (Feng, Nowak, O’Neill, & Welsh, 2014), which is designed for spatio-temporal imputation. We perform a log transformation $\log(x + 1)$ to precipitation level to make its distribution less skew. The goal is then to find a latent representation for the data, interpret the latent components and predict the observed variables to future and to new locations by using iVAE preprocessing. We select nine random stations and all time points from beginning of May 2022 for validation purposes. As a result, we have $n_s = 81$ and $n_t = 874$ in the training data. The validation stations are presented as triangles in Fig. 5 together with the rest of the stations.

In the Veneto region, the elevation of the meteorological stations vary a lot; the maximum altitude is in the north (i.e. Dolomites mountain area) and the minimum altitude can be recognized in the south-east (i.e. Venetian plan). Hence, in this application we consider the spatial dimension $D = 3$, which is composed of the X coordinate, the Y coordinate and the elevation. The locations and the elevations of the stations are presented in Fig. 5. To account for the differences in elevation, we construct radial basis functions also based on the elevation from the sea.

5.1. Interpretation of the latent components

In this section, we use same iVAEr hyperparameters as in Section 4, but we add the radial basis functions based on the elevation. For elevation, we use resolution levels $E = (2, 9)$. We begin with estimating the number of latent components by fitting iVAEr with latent dimensions $P = 3, \dots, 7$ and comparing the ELBOs. For each latent dimension, the method is run for 30 epochs. The ELBOs are presented against different latent dimensions in Fig. 6. The figure shows a clear knee point at $P = 5$ after which the ELBO remains stable. The lowest uAIC is also obtained

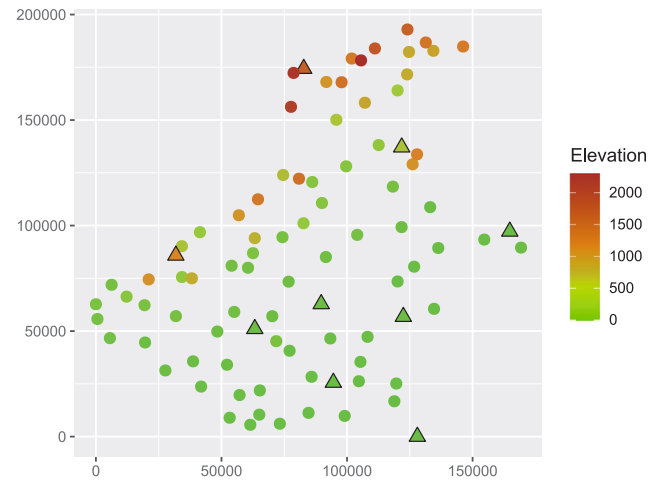


Fig. 5. The spatial locations of meteorological stations in Veneto region. The validation stations are marked as triangles. The colours of the point represent the elevation differences of the stations.

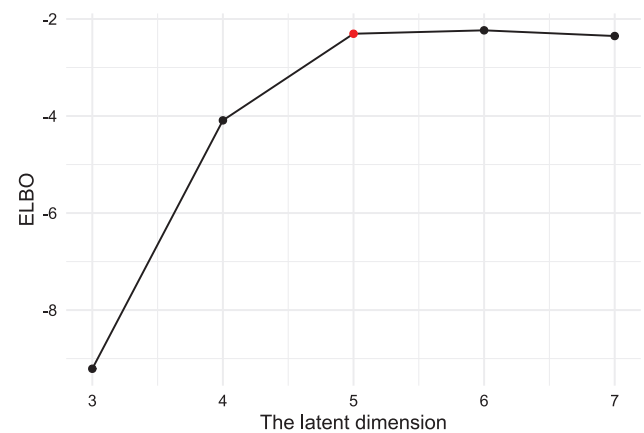


Fig. 6. ELBOs for iVAEr models fitted with the latent dimensions $P = 3, \dots, 7$. The selected latent dimension $P = 5$ is marked as red.

at $P = 5$. Hence, we select the latent dimension $P = 5$ for further analysis.

Next, iVAEr is fitted to the whole dataset by training the model for 60 epochs. To interpret the obtained latent components, we calculate the scaled mean absolute Shapley additive explanations (MASHAP) (Sipilä et al., 2024) for fitted iVAEr’s decoder. The scaled MASHAP are based on Shapley additive explanations (SHAP) (Lundberg & Lee,

Table 3

The scaled MASHAP and the average scaled MASHAP values calculated for iVAEr's decoder after the training process.

| | IC1 | IC2 | IC3 | IC4 | IC5 |
|----------|-------|-------|-------|--------------|--------------|
| ET_0 | 0.125 | 0.026 | 0.029 | 0.721 | 0.099 |
| T_max | 0.037 | 0.036 | 0.090 | 0.722 | 0.115 |
| T_min | 0.148 | 0.029 | 0.095 | 0.659 | 0.069 |
| H_max | 0.170 | 0.026 | 0.484 | 0.037 | 0.284 |
| H_min | 0.129 | 0.045 | 0.280 | 0.160 | 0.386 |
| log_prec | 0.031 | 0.014 | 0.214 | 0.190 | 0.551 |
| wind_vel | 0.038 | 0.798 | 0.106 | 0.029 | 0.028 |
| Average | 0.097 | 0.139 | 0.185 | 0.360 | 0.219 |

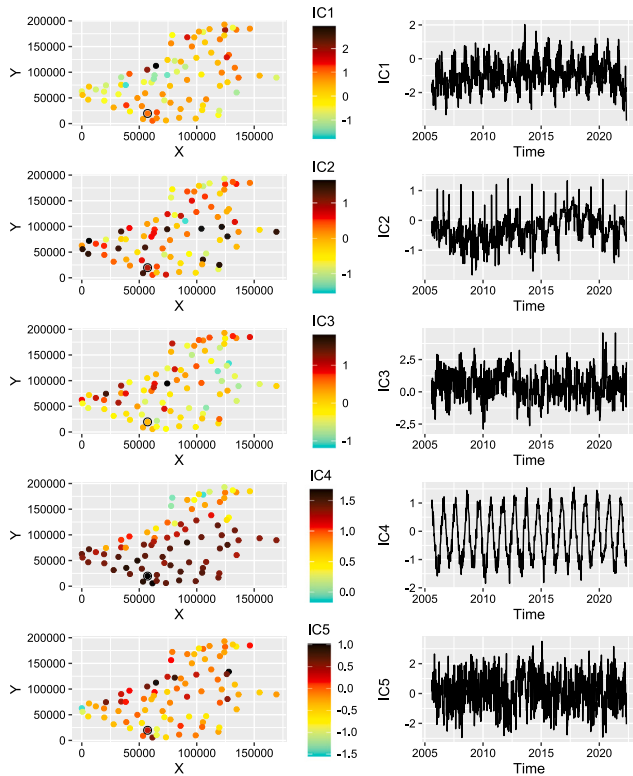


Fig. 7. Spatial map of ICs (left column) for the first time point and a time series of ICs (right column) for the point circled in the spatial maps.

2017) and MASHAP values (Marcilio & Eler, 2020), but modified for obtaining population level feature importances for functions with vector valued output. The scaled MASHAP values can be interpreted as feature importance values for function's input variables, where higher value means that the input variable has more importance for the output variable. By averaging over the scaled MASHAP values for each input variable, the average scaled MASHAP values are obtained, which can be interpreted as population level importances for functions with multiple outputs.

The scaled MASHAP values are calculated for decoder part by using 500 randomly selected observations as background data. The MASHAP values for the decoder part are presented in Table 3. Spatial and temporal behaviours of the latent components are illustrated in Fig. 7. Based on the average scaled MASHAP values, the latent components IC4 and IC5 explain the most of the observed data.

IC1 has small scaled MASHAP values for all variables, which indicates that IC1 is more of a residual component. It still has some seasonal

variability based on the temporal behaviour. IC2 explains only the wind. It has some seasonal variability and high peaks in time irregularly and does not show any clear spatial behaviour. The wind is not highly present in any other components meaning that wind might not share any common latent components with the other variables. IC3 and IC5 together explain the most of the precipitation and humidity. IC3 has the smallest values in east while IC5 has the lowest ones in north-west. Both components seem to have higher values in the mountain area. The components do not have clear trend in time but differently from IC3, IC5 captures high frequency oscillations in time. IC4 captures the seasonal changes as well as elevation based spatial changes. When looking at temporal changes, it has low values during the winter and high values during the summer. Moreover, the low values occur in spatial locations where elevation is high and the high values occur in lowlands. In addition, the coastal area (on the right side of the map) seem to have slightly lower values than inland (the centre and the left side of the map). Evapotranspiration, maximum temperature and minimum temperature have high scaled MASHAP values in IC4 and low values in other ICs, which indicates that IC4 mostly explains these variables. In addition, IC4 explains some of precipitation and minimum humidity.

5.2. Spatio-temporal predictions

In the next, we study if preprocessing the data with iVAEr improves spatio-temporal prediction accuracy. For comparison purposes we use STBSS (Muehlmann et al., 2023) and STLTM approaches (De Iaco, Myers, & Posa, 2003; De Iaco, Palma, & Posa, 2005). In particular, we apply STBSS as preprocessing method and direct predicting using spatio-temporal kriging (Kyriakidis & Journel, 1999), as well as we use the STLTM to forecast the target variables using cokriging (Hans, 2003). We consider four different iVAEr approaches to obtain the latent components z . The approaches are composed of two different iVAEr's auxiliary function settings; one using three hidden layers with 128 units (iVAEaux3) and the other using one hidden layer with 16 units (iVAEaux1). The first two of the approaches use iVAEaux3 and iVAEaux1 directly to predict the latent components to new spatio-temporal locations, and the other two approaches combine iVAEaux3 and iVAEaux1 with spatio-temporal kriging to obtain the predictions. When iVAEaux3 and iVAEaux1 are used directly, the trend function $\mu(s, t)$ provided by the auxiliary function, is treated directly as the predictions for the latent components. When iVAEaux3 and iVAEaux1 are combined with kriging, the latent components $z(s, t)$ are preprocessed further by subtracting the mean function $\mu(s, t)$ as $z_{res}(s, t) = z(s, t) - \mu(s, t)$. The residuals z_{res} are then predicted to new locations using spatio-temporal kriging, and finally, the trend is added back to the predictions as $z(s_{new}, t_{new}) = z_{res}(s_{new}, t_{new}) + \mu(s_{new}, t_{new})$. To obtain the predictions in the original observation space, the predicted latent components are back transformed using iVAEr's mixing function estimate.

For prediction purposes, the auxiliary variable should be formed in a way that the scope of auxiliary variables for the prediction locations is not far out of the scope of the auxiliary variables for the training locations. As our training data has no information of the future time points, the spatio-temporal trend function is not reliable for future time points in regular iVAEr setup. To make the scope same for both training and validation data, we utilize the seasonality of the data instead of using the absolute time points to form the temporal radial basis functions. To account for the seasonality, the temporal radial basis functions are formed using the week of the year (1–53) instead of using the weeks from the first observation (1–892). In addition, to allow the differences between different years, we add a one-hot encoded year factor, i.e. a 18-dimensional standard basis vector giving the year of the observation, to the auxiliary variable. With such setup, the auxiliary variables used for predictions are not out of scope of the variables used for training and hence, iVAEr's auxiliary function is able to predict

the trend to the future better than by using the regular setup with temporal radial basis functions based on the absolute time points. The auxiliary function of iVAEaux3 with three hidden layers has capability of learning finer details of the spatio-temporal field, but is also more prone to overfitting. The auxiliary function of iVAEaux1 on the other hand is less likely to overfit as it has much less parameters and might be for that reason better for trend estimation. In each iVAE setting, we use the spatial resolution levels $H = (2, 9)$, temporal (seasonal) resolution levels $G = (2, 9)$ and elevation based resolution levels $E = (2, 9)$.

The reference methods, direct spatio-temporal kriging, STBSS combined with kriging and STLTCM with cokriging, expect the data without seasonal components. Since the weather data has clear seasonality, we hence preprocess the data by subtracting the estimated seasonal component from each original variable. The seasonal cyclicity is estimated by fitting the model

$$x_i(s, t) = \beta_{0,i} + \beta_{1,i}\cos(2\pi t/53) + \beta_{2,i}\sin(2\pi t/53) + x_{res,i}(s, t)$$

for each variable x_i , $i = 1, \dots, 7$. In direct kriging approach, the residuals $x_{res,i}$ are predicted using spatio-temporal kriging. In STBSS based approach, STBSS is applied to the residuals to obtain the latent components. The latent components are then predicted using kriging, and the predictions are backtransformed to original data using STBSS. The parameters for STBSS are the best performing parameters based on the simulation studies, but scaled to match the spatial domain of Veneto dataset. In the STLTCM, we first have selected 3 latent components identified by joint diagonalization of the sample covariance matrices (Cappello, De Iaco, & Palma, 2022; De Iaco, Myers, Palma, & Posa, 2013; De Iaco, Palma & Posa, 2019; De Iaco & Posa, 2012). Then, supported by the evaluation of the non-separability index (Cappello, De Iaco, & Posa, 2020; De Iaco & Posa, 2013; De Iaco, Posa, Cappello & Maggio, 2019) the product-sum covariance model has been chosen to describe the characteristics of the covariance functions estimated on the three retained components. Finally, the STLTCM model has been used to compute spatio-temporal predictions of the seven analysed variables.

To estimate the seasonal components $\mu_i(s, t) = \hat{\beta}_{0,i} + \hat{\beta}_{1,i}\cos(2\pi t/53) + \hat{\beta}_{2,i}\sin(2\pi t/53)$ for new spatial locations we use the modified version of the approach used in De Iaco and Posa (2012) for each variable x_i , $i = 1, \dots, 7$:

1. Calculate means $\mu_i(s_j)$ and standard deviations $\sigma_i(s_j)$ of the seasonal components for the training stations s_j , $j = 1, \dots, n_s$. Standardize the seasonal components for each training station to have zero mean and unit variance as $\tilde{\mu}_i(s_j, t) = \frac{\mu_i(s_j, t) - \mu_i(s_j)}{\sigma_i(s_j)}$.
2. Fit the periodic function $\tilde{\mu}_i(s_j, t) = \tilde{\beta}_{1,i}\cos(2\pi t/53) + \tilde{\beta}_{2,i}\sin(2\pi t/53) + \tilde{\mu}_{res,i}(s_j, t)$ for the standardized seasonal components to obtain the periodic component $\tilde{\mu}_i(t) = \tilde{\beta}_{1,i}\cos(2\pi t/53) + \tilde{\beta}_{2,i}\sin(2\pi t/53)$.
3. Consider the means $\mu_i(s_j)$ and standard deviations $\sigma_i(s_j)$ as realizations of two spatial random fields. Use spatial kriging to estimate the mean and the variance to new locations s_{new} .
4. Construct the trend to location s_{new} as $\mu_i(s_{new}, t) = \tilde{\mu}_i(t)\sigma_i(s_{new}) + \mu_i(s_{new})$.

The final predictions $x_i(s_{new}, t_{new})$ are then obtained as $x_i(s_{new}, t_{new}) = x_{res,i}(s_{new}, t_{new}) + \mu_i(s_{new}, t_{new})$, where $x_{res,i}(s_{new}, t_{new})$ is the predicted residual of the deseasonalized observation. In kriging step for both original variables and for latent components provided by iVAER and STBSS, we fit either integrated product sum covariance model (De Iaco, Myers, & Posa, 2002) or the product sum covariance model (De Iaco, Myers, & Posa, 2001) based on the non-separability index. The kriging predictions are calculated using 40 nearest points.

As measures of performance, we calculate mean squared error $MSE(x_i, \hat{x}_i) = \frac{1}{n} \sum_{j=1}^n (x_{i,j} - \hat{x}_{i,j})^2$ and mean absolute error $MAE(x_i, \hat{x}_i) = \frac{1}{n} \sum_{j=1}^n |x_{i,j} - \hat{x}_{i,j}|$, where x_i and \hat{x}_i has the true observations and

Table 4

Mean squared errors of iVAE with three hidden layers in auxiliary function (iVAEaux3), iVAEaux3 combined with kriging, iVAE with one hidden layer in auxiliary function (iVAEaux1), iVAEaux1 combined with kriging, regular univariate spatio-temporal kriging, STBSS combined with kriging and STLTCM combined with cokriging. The smallest errors are bolded for each variable.

| (a) MSEs and wMSE for temporal part | | | | | | | |
|-------------------------------------|-------------|-------------|-------------|--------------|---------------|-------------|--------------|
| | ET_0 | T_max | T_min | H_max | H_min | log_prec | wMSE |
| iVAEaux3 | 0.06 | 3.72 | 4.21 | 87.28 | 175.69 | 2.12 | 18.15 |
| iVAEaux3 + kriging | 0.07 | 4.47 | 4.29 | 85.00 | 165.08 | 2.15 | 16.20 |
| iVAEaux1 | 0.07 | 4.98 | 5.24 | 117.92 | 173.91 | 2.58 | 22.83 |
| iVAEaux1 + kriging | 0.11 | 7.90 | 5.21 | 115.73 | 235.22 | 2.02 | 20.59 |
| Kriging | 0.12 | 6.28 | 3.49 | 89.37 | 358.14 | 1.97 | 19.38 |
| STBSS + kriging | 0.11 | 8.04 | 3.51 | 91.69 | 335.65 | 1.95 | 16.70 |
| STLTCM + cokriging | 0.08 | 5.55 | 3.84 | 79.66 | 207.99 | 1.96 | 17.83 |

| (b) MSEs and wMSE for spatial part | | | | | | | |
|------------------------------------|-------------|-------------|-------------|--------------|--------------|-------------|--------------|
| | ET_0 | T_max | T_min | H_max | H_min | log_prec | wMSE |
| iVAEaux3 | 0.10 | 5.33 | 6.61 | 74.45 | 123.91 | 1.84 | 104.05 |
| iVAEaux3 + kriging | 0.07 | 2.27 | 4.00 | 52.43 | 44.78 | 0.28 | 125.16 |
| iVAEaux1 | 0.12 | 6.72 | 9.11 | 65.07 | 141.78 | 2.19 | 41.73 |
| iVAEaux1 + kriging | 0.05 | 1.55 | 3.22 | 33.82 | 33.14 | 0.25 | 38.34 |
| Kriging | 0.19 | 8.22 | 2.47 | 37.90 | 57.66 | 0.26 | 39.06 |
| STBSS + kriging | 0.19 | 8.30 | 2.49 | 36.52 | 59.76 | 0.30 | 39.00 |
| STLTCM + cokriging | 0.17 | 8.20 | 2.42 | 37.76 | 56.74 | 0.23 | 39.02 |

| (c) MSEs and wMSE for spatio-temporal part | | | | | | | |
|--|-------------|-------------|-------------|---------------|---------------|-------------|--------------|
| | ET_0 | T_max | T_min | H_max | H_min | log_prec | wMSE |
| iVAEaux3 | 0.07 | 3.65 | 6.64 | 249.94 | 258.02 | 2.37 | 106.01 |
| iVAEaux3 + kriging | 0.07 | 3.57 | 6.98 | 249.96 | 251.24 | 2.42 | 123.50 |
| iVAEaux1 | 0.07 | 5.47 | 7.24 | 208.56 | 206.50 | 2.53 | 35.80 |
| iVAEaux1 + kriging | 0.09 | 6.57 | 6.62 | 187.59 | 177.25 | 2.25 | 33.83 |
| Kriging | 0.28 | 14.40 | 6.57 | 207.04 | 409.58 | 2.13 | 58.01 |
| STBSS + kriging | 0.20 | 12.98 | 5.88 | 199.71 | 334.76 | 2.02 | 32.43 |
| STLTCM + cokriging | 0.18 | 10.41 | 6.06 | 184.22 | 251.47 | 1.94 | 33.82 |

predicted counterparts of i th observed variable, respectively. In addition, we calculate weighted average MSE (wMSE) and weighted average MAE (wMAE), which are calculated as $wMSE(X, \hat{X}) = \frac{1}{7} \sum_{i=1}^7 \frac{MSE(x_i, \hat{x}_i)}{\sigma^2(x_i)}$ and $wMAE(X, \hat{X}) = \frac{1}{7} \sum_{i=1}^7 \frac{MAE(x_i, \hat{x}_i)}{\sigma(x_i)}$, where X is a $n \times 7$ matrix containing the true observation vectors as rows, \hat{X} has the predicted observations and $\sigma(x_i)$ is a standard deviation of i th observed variable calculated from the deseasonalized train data. wMSE and wMAE account for the scale differences of the variables and give a single measure of performance regarding the whole data.

The validation prediction errors are collected in Tables 4 and 5, where the cases (a), (b) and (c) has the prediction errors for temporal part, spatial part and spatio-temporal part, respectively. When the observed variables are predicted to future (cases (a) in Tables 4 and 5) considering only the spatial locations present in the training data, the best overall results are obtained using iVAEaux3 or iVAEaux3 + kriging. They have slightly lower wMSE and wMAE than the second best method, STLTCM + cokriging, and clearly outperform the other methods. When the prediction errors of the individual variables are inspected, it is evident that iVAEaux3 and iVAEaux3 + kriging provide the lowest MSE values for all but minimum temperature, maximum humidity and log precipitation. The lowest MSE for these variables are obtained by kriging, STLTCM + cokriging and STBSS + kriging, respectively. The lowest MAE for all variables but minimum temperature and log precipitation are also obtained by iVAEaux3 and iVAEaux3 + kriging. The lowest MAE for minimum temperature and log precipitation are obtained by STBSS + kriging and iVAEaux1 + kriging, respectively.

When the observed variables are predicted to new spatial locations (cases (b) in Tables 4 and 5), but not to the future (i.e. using all time points from the training data), the best overall results are obtained by iVAEaux1 + kriging as it has significantly lower wMSE and wMAE than any of the competing methods. iVAEaux1 + kriging has the lowest MAE

Table 5

Mean absolute errors of iVAE with three hidden layers in auxiliary function (iVAEaux3), iVAEaux3 combined with kriging, iVAE with one hidden layer in auxiliary function (iVAEaux1), iVAEaux1 combined with kriging, regular univariate spatio-temporal kriging, STBSS combined with kriging and STLTCM combined with cokriging. The smallest errors are bolded for each variable.

| (a) MAEs and wMAE for temporal part | | | | | | | | |
|-------------------------------------|-------------|-------------|-------------|-------------|--------------|-------------|-------------|-------------|
| | ET_0 | T_max | T_min | H_max | H_min | log_prec | wind_vel | wMAE |
| iVAEaux3 | 0.19 | 1.63 | 1.65 | 6.05 | 10.73 | 1.22 | 2.77 | 0.76 |
| iVAEaux3 + kriging | 0.20 | 1.75 | 1.68 | 5.99 | 10.41 | 1.25 | 2.55 | 0.76 |
| iVAEaux1 | 0.21 | 1.84 | 1.78 | 7.04 | 10.77 | 1.45 | 3.52 | 0.86 |
| iVAEaux1 + kriging | 0.25 | 2.21 | 1.85 | 7.04 | 12.34 | 1.17 | 3.06 | 0.88 |
| Kriging | 0.27 | 2.06 | 1.54 | 6.11 | 15.67 | 1.26 | 3.10 | 0.88 |
| STBSS + kriging | 0.25 | 2.34 | 1.52 | 6.22 | 14.98 | 1.23 | 2.71 | 0.87 |
| STLTCM + cokriging | 0.24 | 1.94 | 1.60 | 6.09 | 11.66 | 1.26 | 2.86 | 0.81 |

| (b) MAEs and wMAE for spatial part | | | | | | | | |
|------------------------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| | ET_0 | T_max | T_min | H_max | H_min | log_prec | wind_vel | wMAE |
| iVAEaux3 | 0.24 | 1.80 | 2.04 | 5.51 | 8.35 | 1.10 | 7.63 | 0.96 |
| iVAEaux3 + kriging | 0.20 | 1.14 | 1.49 | 4.83 | 4.49 | 0.37 | 8.36 | 0.77 |
| iVAEaux1 | 0.25 | 2.05 | 2.39 | 5.06 | 9.33 | 1.25 | 5.19 | 0.91 |
| iVAEaux1 + kriging | 0.16 | 0.96 | 1.28 | 3.69 | 4.18 | 0.33 | 4.93 | 0.56 |
| Kriging | 0.27 | 1.98 | 1.17 | 4.19 | 5.78 | 0.36 | 5.16 | 0.70 |
| STBSS + kriging | 0.28 | 1.99 | 1.18 | 4.15 | 5.86 | 0.41 | 5.16 | 0.70 |
| STLTCM + cokriging | 0.26 | 1.89 | 1.17 | 4.17 | 5.79 | 0.30 | 5.11 | 0.68 |

| (c) MAEs and wMAE for spatio-temporal part | | | | | | | | |
|--|-------------|-------------|-------------|-------------|--------------|-------------|-------------|-------------|
| | ET_0 | T_max | T_min | H_max | H_min | log_prec | wind_vel | wMAE |
| iVAEaux3 | 0.20 | 1.59 | 2.19 | 10.92 | 12.58 | 1.31 | 7.59 | 1.14 |
| iVAEaux3 + kriging | 0.20 | 1.59 | 2.26 | 10.91 | 12.44 | 1.34 | 8.27 | 1.17 |
| iVAEaux1 | 0.22 | 1.89 | 2.05 | 9.70 | 11.68 | 1.43 | 4.21 | 0.99 |
| iVAEaux1 + kriging | 0.23 | 2.10 | 2.05 | 9.36 | 10.96 | 1.28 | 4.26 | 0.97 |
| Kriging | 0.36 | 3.04 | 2.15 | 9.71 | 16.70 | 1.30 | 6.12 | 1.24 |
| STBSS + kriging | 0.32 | 2.92 | 1.99 | 9.85 | 15.11 | 1.29 | 4.50 | 1.12 |
| STLTCM + cokriging | 0.32 | 2.43 | 2.07 | 9.78 | 12.67 | 1.26 | 4.67 | 1.07 |

and MSE for evapotranspiration, maximum temperature, maximum humidity and minimum humidity and wind velocity. The lowest MSE and MAE for minimum temperature and log precipitation are obtained by STLTCM + cokriging.

When the observed variables are predicted to future and to new spatial locations (cases (c) in Tables 4 and 5), iVAEaux1 + kriging is the best performing method, followed by iVAEaux1 and STLTCM + cokriging. The lowest wMSE and wMAE are both obtained by iVAEaux1 + kriging. However, when inspecting MSE and MAE values for individual variables, the results are spread more between the methods. All iVAE based methods have the lowest MSE and MAE values for evapotranspiration. For maximum temperature the differences in the errors are high. The lowest errors are obtained by iVAEaux3 based methods, followed by iVAEaux1 based methods, while for minimum temperature, STBSS + kriging has the lowest value, but the differences between the errors are low. The lowest MSE for maximum humidity is obtained by STLTCM + cokriging and the lowest MAE by iVAEaux1 + kriging. MSE and MAE for minimum humidity are clearly lowest for iVAEaux1 + kriging. For log precipitation and wind velocity, the differences between the best performing methods small. The lowest errors for log precipitation are obtained by STLTCM + cokriging, and for wind velocity the lowest MSE is obtained by STBSS + kriging and the lowest MAE by iVAEaux1.

In conclusion, it is evident that iVAEr, estimating the latent components and spatio-temporal trend (and variance) function simultaneously, clearly improves the prediction accuracy compared to competing methods, especially in spatial data case. For temporal prediction, the best accuracy is obtained by iVAEaux3 using the auxiliary function with three hidden layers. This hints that for temporal prediction, it is beneficial to capture finer details of the spatio-temporal structure. For temporal part, kriging did not improve significantly the prediction accuracy meaning that it is sufficient to use solely iVAEr in this situation.

For spatial prediction, in the other hand, iVAEaux1, using one hidden layer with only 16 neurons in auxiliary function, provides better results than iVAEaux3 hinting that it is more beneficial to capture only larger scale spatio-temporal structure and predict the residuals further using spatio-temporal kriging. In addition to performance gains of iVAEr based methods, they benefit from more simple modelling process as there is no need for fitting and predicting the spatio-temporal trend function separately as this is done simultaneously by the algorithm. Moreover, by providing independent components, the prediction can be done efficiently using univariate prediction methods such as kriging, which is computationally more manageable as compared to multivariate prediction methods such as cokriging. Although even in this last approach, the multivariate estimation is often simplified by modelling uncorrelated components, identified through the joint diagonalization of the covariance matrices.

6. Conclusion and discussion

In this paper, iVAE was extended to the nonstationary spatio-temporal setting, and three approaches, coordinate based, segmentation based and radial basis function based, were introduced for constructing the spatio-temporal auxiliary data. In addition, two latent dimension estimation methods were proposed. The introduced spatio-temporal iVAE and latent dimension estimation methods were studied using vast simulation studies and illustrated in meteorological application where also a novel iVAEr preprocessing procedure for accounting nonstationarity in spatio-temporal modelling was introduced.

Based on the simulations, iVAEr, iVAEs1 and iVAEs2 were the best performing methods in nonlinear STBSS settings with nonstationary spatio-temporal variance. The methods outperformed iVAEc, iVAEs3 and STBSS methods in all settings. FICA was still the best performing method under the linear mixing in settings without spatio-temporal trend and highly nonlinear variance. However, under nonlinear mixing or when the trend was present, iVAE methods outperformed FICA. Based on the fact that iVAEr provides for the latent components smooth trend and variance function estimates, which can be useful for further analysis, we consider iVAEr the best method for nonstationary STBSS.

In meteorological application, we utilized the introduced latent dimension estimation methods and iVAEr to find the underlying latent components. We interpreted the components using scaled MASHAP values and by inspecting the spatial and temporal behaviours of the components. Original seven variables were compressed into five latent components; one explained the seasonal and spatial variability in temperature and evapotranspiration, one explained the wind velocity, two explained together the most of the precipitation and humidity and the last one explained the remaining residuals of the data. For spatio-temporal prediction purposes, we utilized iVAEr preprocessing by estimating the latent components and their nonstationary spatio-temporal trend functions, which were used to account nonstationarity in the modelling of the latent components. When using iVAEr preprocessing, the prediction accuracy was improved as compared to predicting the original variables directly (through kriging or cokriging) or to using STBSS as preprocessing method.

Based on the results and theoretical properties of the developed spatio-temporal iVAE methods, it is evident that the methods have multiple advantages over the previously proposed STBSS methods. The methods are capable of estimating nonlinear mixing and unmixing functions in nonstationary data settings, whereas the previously proposed methods are designed only for linear mixing and stationary data settings. The developed iVAE methods can estimate injective mixing function, while previous linear methods assume that the latent and the observed dimensions are equal. In addition, iVAEr is highly useful for spatio-temporal modelling and prediction as it can estimate nonstationary spatio-temporal trend and variance functions. By removing the nonstationary trend from the components, stationary prediction

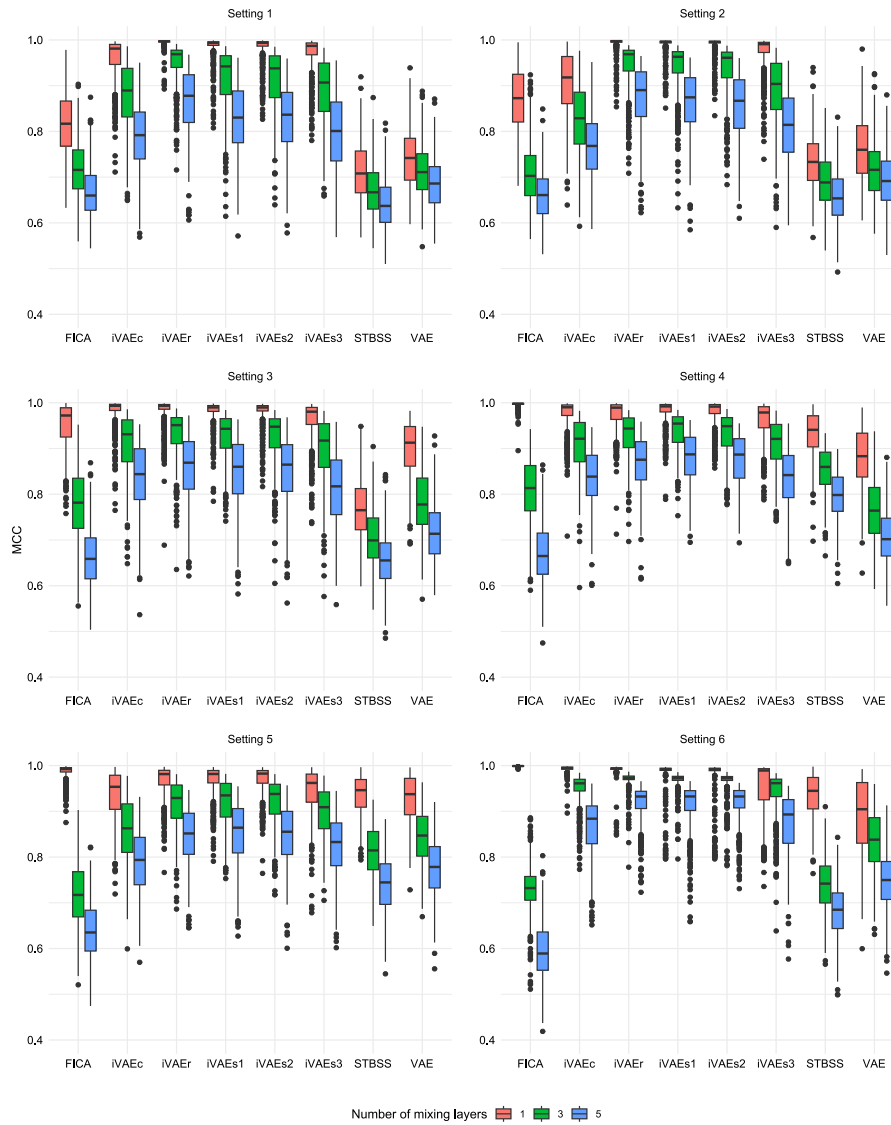


Fig. B.8. Mean correlation coefficients of 500 trials for Settings 1–6 for sample size with the number of spatial locations $n_s = 50$ and the number of temporal observations $n_t = 300$.

methods, such as kriging, can provide better predictions as seen in the meteorological application.

Although in this paper we performed a small simulation study regarding sensitivity of iVAEr against different radial basis function settings, more in depth study of sensitivity against different hyper-parameters or source density mismatch is needed and will be done in future. The spatio-temporal iVAE methods developed in this paper rely on nonstationary variance for identifiability. As discussed in Section 4.1, there are other possibilities for introducing nonstationarity in spatio-temporal settings. The methods for other scenarios, such as nonstationary autocorrelation, will be developed in the future. Also, nonlinear STBSS methods for stationary data are left for future work. Nonlinear SBSS and STBSS methods have so far been studied and developed mainly for Gaussian data. In future, different source densities will be considered and methods that are resistant to outliers will be developed in a similar manner as was done in linear BSS framework in Sipilä, Muehlmann, Nordhausen and Taskinen (2024). In future, we will also focus on developing nonlinear BSS methods for graph data,

where one cannot measure the distance between the observations as in temporal, spatial or spatio-temporal framework, but only the relations between the observations are known.

Although this paper focuses exclusively on a meteorological application, the spatio-temporal iVAE methods are applicable to any multivariate spatio-temporal data. For instance, in neuroimaging, magnetic resonance spectroscopic imaging (MRSI) measures multiple chemicals across the brain over a period of time, making it a multivariate spatio-temporal application. Similarly, many traditional neuroimaging techniques such as electroencephalography (EEG) or magnetoencephalography (MEG) can be viewed as univariate spatio-temporal data. If multiple similar clinical trials are performed on the same patient, the resulting data can be treated as multivariate spatio-temporal.

In such scenarios, spatio-temporal iVAE methods could uncover latent spatio-temporal patterns in brain function that are not directly observable. To further explore the versatility of the proposed iVAE methods, future work should include diverse real-world data examples, such as neuroimaging or epidemiological pattern studies.

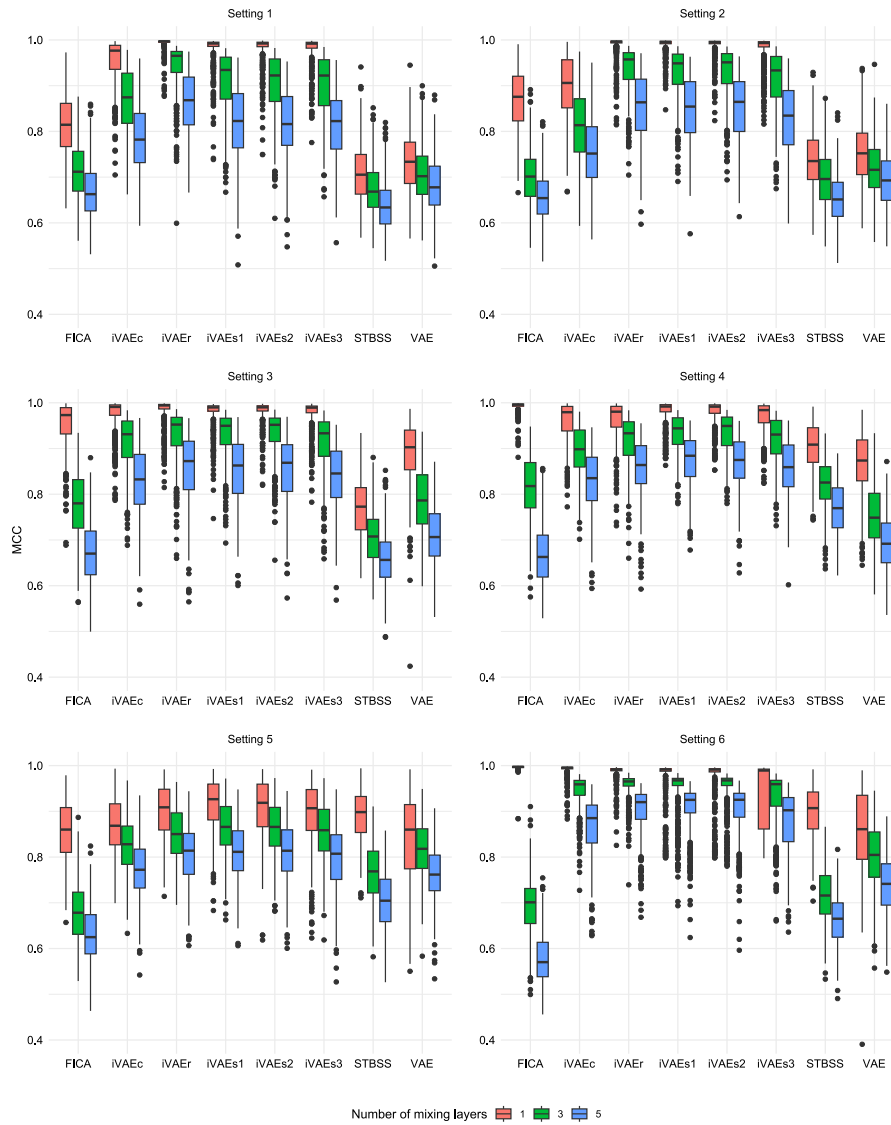


Fig. B.9. Mean correlation coefficients of 500 trials for settings 1–6 for sample size with the number of spatial locations $n_s = 150$ and the number of temporal observations $n_t = 75$.

CRedit authorship contribution statement

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

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.

Data availability

Data will be made available on request.

Acknowledgements

We acknowledge the support from Vilho, Yrjö and Kalle Väisälä foundation for MS, the support from the Research Council of Finland (453691) to ST, the support from the Research Council of Finland (363261) to KN and the support from the HiTEc COST Action (CA21163) to KN and ST.

Appendix A. Computational complexity analysis

The computational complexities of the proposed algorithms are composed of two parts, forming the auxiliary data and training iVAE. We use Big O notation to represent the worst case time and space complexities, where $O(n)$ denotes linear growth in computation time or memory usage with respect to the input size n . First, let us address iVAE's computational complexity which is very similar to any feed forward neural network such as regular VAE. Using the Big O notation, the computational time complexity for training the model is $O(n \times n_w \times n_e)$, where n is the sample size, n_w is the number of weights in the model and n_e is the number of epochs. When the sample size n grows, less epochs are typically needed for training, which makes the model well scaleable

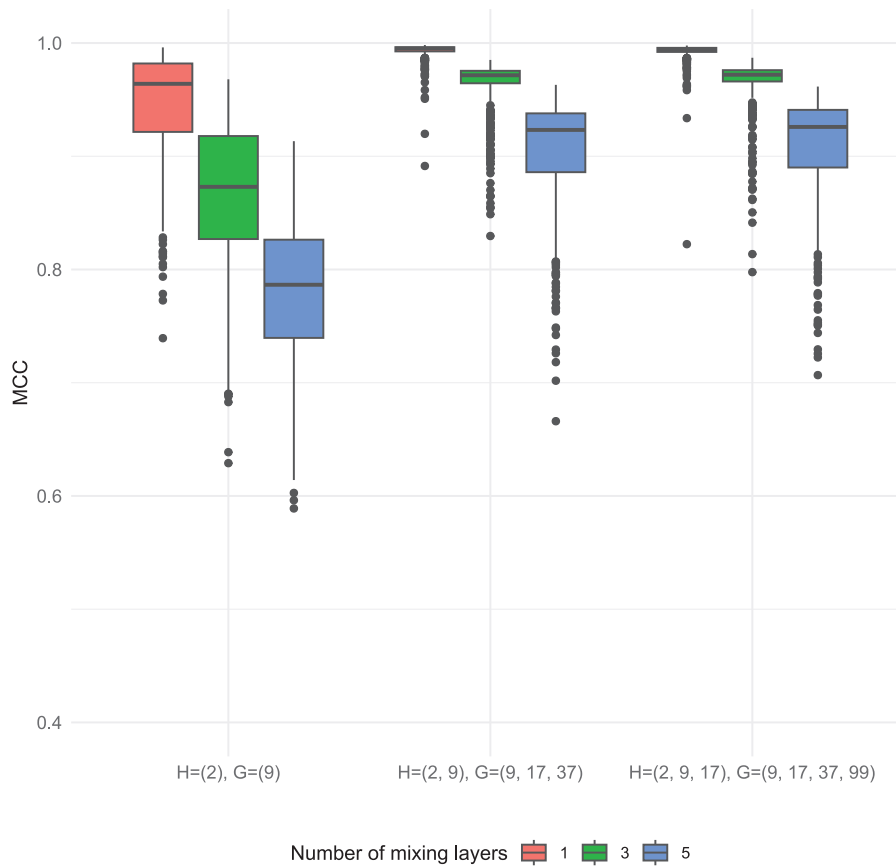


Fig. B.10. Mean correlation coefficients for different radial basis function parameter settings of iVAEr. The boxplots present 500 trials for Setting 6 with the number of spatial locations $n_s = 150$ and the number of temporal observations $n_t = 300$.

in terms of sample size. The memory usage, *i.e.* space complexity, is $O(n_w)$ for storing the weights of the model. The number of weights n_w can be broken down to number of weights n_{w1} in encoder-decoder part and the number of weights n_{w2} in auxiliary function. n_{w1} and n_{w2} are heavily dependent on the number and size of the hidden layers. Typically fairly small neural networks (*e.g.* 3 layers with 128 units) in encoder, decoder and auxiliary functions are sufficient. In addition, n_{w1} depends linearly on the dimension of the input data and latent dimension whereas n_{w2} depends linearly on the dimension of auxiliary data. Also, if the input dimension is very large (*e.g.* more than 100), a larger encoder-decoder network might be needed. Hence, the time complexity grows more when input dimension, latent dimension or auxiliary data dimension grow.

In iVAEc, the time and space complexities are the lowest as the coordinates are only scaled to form the auxiliary variables, meaning that using Big O notation, both time and space complexities are $O(n)$ for forming and storing the auxiliary data. The auxiliary data is only two dimensional, which makes time complexity slightly lower compared to other algorithms. In iVAEs1-iVAEs3, the time complexity of forming the auxiliary data is $O(m_{S_1} \times m_{S_2} \times m_{\mathcal{T}} \times n)$, where m_{S_1} , m_{S_2} and $m_{\mathcal{T}}$ are the number of segments along each dimension. However, for iVAEs1, where all dimensions are considered jointly, and hence the dimension of auxiliary data can be very large, the space complexity is $O(m_{S_1} \times m_{S_2} \times m_{\mathcal{T}} \times n)$. For iVAEs2, the space complexity is $O((m_{S_1} \times m_{S_2} + m_{\mathcal{T}}) \times n)$ and for iVAEs3, it is $O((m_{S_1} + m_{S_2} + m_{\mathcal{T}}) \times n)$. In terms of computation time, iVAEs3 is the most efficient of segmentation based algorithms as the auxiliary dimension is the lowest. iVAEs2 is also efficient if the number of spatial segments is not very high. In iVAEr, the time and space complexities for forming and storing auxiliary data are $O(K_S + K_{\mathcal{T}}) \times n$, where K_S and $K_{\mathcal{T}}$ are numbers of spatial and temporal node points, respectively. In all above iVAE variants, the space complexity

can be reduced further by constructing auxiliary variables batch-wise during training process. In conclusion, the algorithms are well scalable in terms of sample size n and relatively well scalable in terms of dimensions of input data, latent data and auxiliary data (linear time complexity). However, if the auxiliary data are not formed batch-wise, memory consumption may grow large if dimension of auxiliary variable is very large. Since the algorithm is essentially composed of three feed forward neural networks, encoder, decoder and auxiliary function, standard parallelization methods such as data parallelism, which distributes the data batch-wise across multiple computation units, can be applied to further reduce the overall computation time.

Appendix B. Additional simulation results

See Figs. B.8–B.10

References

- Ansari, S., Alatrany, A. S., Alnajjar, K. A., Khater, T., Mahmoud, S., Al-Jumeily, D., et al. (2023). A survey of artificial intelligence approaches in blind source separation. *Neurocomputing*, 561, Article 126895. <http://dx.doi.org/10.1016/j.neucom.2023.126895>.
- Bachoc, F., Genton, M. G., Nordhausen, K., Ruiz-Gazen, A., & Virta, J. (2020). Spatial blind source separation. *Biometrika*, 107, 627–646. <http://dx.doi.org/10.1093/biomet/asz079>.
- Cappello, C., De Iaco, S., & Palma, M. (2022). Computational advances for spatio-temporal multivariate environmental models. *Computational Statistics*, 37(2), 651–670.
- Cappello, C., De Iaco, S., & Posa, D. (2020). Covatest: an R package for selecting a class of space-time covariance functions. *Journal of Statistical Software*, 94, 1–42.
- Chen, W., Genton, M. G., & Sun, Y. (2021). Space-time covariance structures and models. *Annual Review of Statistics and Its Application*, 8, 191–215.
- Chen, W., Li, Y., Reich, B. J., & Sun, Y. (2024). Deepkriging: Spatially dependent deep neural networks for spatial prediction. *Statistica Sinica*, 34, 291–311.

- Comon, P., & Jutten, C. (2010). Handbook of blind source separation: Independent component analysis and applications. Academic Press, <http://dx.doi.org/10.1016/C2009-0-19334-0>.
- De Iaco, S., Myers, D., Palma, M., & Posa, D. (2013). Using simultaneous diagonalization to identify a space–time linear coregionalization model. *Mathematical Geosciences*, 45, 69–86.
- De Iaco, S., Myers, D. E., & Posa, D. (2001). Space–time analysis using a general product–sum model. *Statistics & Probability Letters*, 52(1), 21–28.
- De Iaco, S., Myers, D. E., & Posa, D. (2002). Nonseparable space–time covariance models: Some parametric families. *Mathematical Geology*, 34, 23–42.
- De Iaco, S., Myers, D., & Posa, D. (2003). The linear coregionalization model and the product–sum space–time variogram. *Mathematical Geology*, 35, 25–38.
- De Iaco, S., Palma, M., & Posa, D. (2005). Modeling and prediction of multivariate space–time random fields. *Computational Statistics & Data Analysis*, 48(3), 525–547.
- De Iaco, S., Palma, M., & Posa, D. (2019). Choosing suitable linear coregionalization models for spatio-temporal data. *Stochastic Environmental Research and Risk Assessment*, 33, 1419–1434.
- De Iaco, S., & Posa, D. (2012). Predicting spatio-temporal random fields: some computational aspects. *Computational Geosciences*, 41, 12–24.
- De Iaco, S., & Posa, D. (2013). Positive and negative non-separability for space–time covariance models. *Journal of Statistical Planning and Inference*, 143(2), 378–391.
- De Iaco, S., Posa, D., Cappello, C., & Maggio, S. (2019). Isotropy, symmetry, separability and strict positive definiteness for covariance functions: a critical review. *Spatial Statistics*, 29, 89–108.
- Feng, L., Nowak, G., O'Neill, T., & Welsh, A. (2014). CUTOFF: A spatio-temporal imputation method. *Journal of Hydrology*, 519, 3591–3605.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2020). Generative adversarial networks. *Communications of the ACM*, 63(11), 139–144.
- Hälvä, H., & Hyvärinen, A. (2020). Hidden Markov nonlinear ICA: Unsupervised learning from nonstationary time series. In *Conference on Uncertainty in Artificial Intelligence* (pp. 939–948). PMLR.
- Hälvä, H., Le Corff, S., Lehericy, L., So, J., Zhu, Y., Gassiat, E., et al. (2021). Disentangling identifiable features from noisy data with structured nonlinear ICA. *Advances in Neural Information Processing Systems*, 34, 1624–1633.
- Hans, W. (2003). *Multivariate Geostatistics: An Introduction with Applications*. Springer Science & Business Media.
- Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). vol. 2, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.
- Hyvärinen, A. (1999). Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3), 626–634.
- Hyvärinen, A., & Morioka, H. (2016). Unsupervised feature extraction by time-contrastive learning and nonlinear ICA. *Advances in Neural Information Processing Systems*, 29.
- Hyvärinen, A., & Morioka, H. (2017). Nonlinear ICA of temporally dependent stationary sources. In *Artificial intelligence and statistics* (pp. 460–469). PMLR.
- Hyvärinen, A., & Pajunen, P. (1999). Nonlinear independent component analysis: Existence and uniqueness results. *Neural Networks*, 12(3), 429–439.
- Hyvärinen, A., Sasaki, H., & Turner, R. (2019). Nonlinear ICA using auxiliary variables and generalized contrastive learning. In *The 22nd international conference on artificial intelligence and statistics* (pp. 859–868). PMLR.
- Khemakhem, I., Kingma, D., Monti, R., & Hyvärinen, A. (2020). Variational autoencoders and nonlinear ICA: A unifying framework. In *International Conference on Artificial Intelligence and Statistics* (pp. 2207–2217). PMLR.
- Kingma, D. P., & Welling, M. (2013). Auto-encoding variational Bayes. arXiv preprint arXiv:1312.6114.
- Kyriakidis, P. C., & Journel, A. G. (1999). Geostatistical space–time models: A review. *Mathematical Geology*, 31, 651–684.
- Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30.
- Luo, W., & Li, B. (2016). Combining eigenvalues and variation of eigenvectors for order determination. *Biometrika*, 103, 875–887.
- Luo, W., & Li, B. (2021). On order determination by predictor augmentation. *Biometrika*, 108, 557–574.
- Maas, A. L., Hannun, A. Y., Ng, A. Y., et al. (2013). Rectifier nonlinearities improve neural network acoustic models. 30, In *Proc. icml* (1), (p. 3). Atlanta, GA.
- Marchini, J. L., Heaton, C., & Ripley, B. D. (2021). FastICA: FastICA algorithms to perform ICA and projection pursuit. URL <https://CRAN.R-project.org/package=fastICA>.
- Marcílio, W. E., & Eler, D. M. (2020). From explanations to feature selection: assessing SHAP values as feature selection mechanism. In *2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)* (pp. 340–347). Ieee.
- Muehlmann, C., Bachoc, F., & Nordhausen, K. (2022). Blind source separation for non-stationary random fields. *Spatial Statistics*, 47, Article 100574. <http://dx.doi.org/10.1016/j.spasta.2021.100574>.
- Muehlmann, C., Bachoc, F., Nordhausen, K., & Yi, M. (2024). Test of the latent dimension of a spatial blind source separation model. *Statistica Sinica*, 34, 837–865. <http://dx.doi.org/10.5705/ss.202021.0326>.
- Muehlmann, C., De Iaco, S., & Nordhausen, K. (2023). Blind recovery of sources for multivariate space-time random fields. *Stochastic Environmental Research and Risk Assessment*, 37, 1593–1613. <http://dx.doi.org/10.1007/s00477-022-02348-2>.
- Muehlmann, C., Piccolotto, N., Cappello, C., De Iaco, S., & Nordhausen, K. (2022). SpaceTimeBSS: Blind source separation for multivariate spatio-temporal data. URL <https://CRAN.R-project.org/package=SpaceTimeBSS>.
- Nag, P., Sun, Y., & Reich, B. J. (2023). Spatio-temporal DeepKriging for interpolation and probabilistic forecasting. *Spatial Statistics*, 57, Article 100773. <http://dx.doi.org/10.1016/j.spasta.2023.100773>.
- Nordhausen, K., Oja, H., Filzmoser, P., & Reimann, C. (2015). Blind source separation for spatial compositional data. *Mathematical Geosciences*, 47(7), 753–770. <http://dx.doi.org/10.1007/s11004-014-9559-5>.
- Nordhausen, K., Taskinen, S., & Virta, J. (2022). Signal dimension estimation in BSS models with serial dependence. In *2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)* (pp. 1–7).
- Nychka, D., Bandyopadhyay, S., Hammerling, D., Lindgren, F., & Sain, S. (2015). A multiresolution Gaussian process model for the analysis of large spatial datasets. *Journal of Computational and Graphical Statistics*, 24(2), 579–599.
- Papalexiou, S. M., & Serinaldi, F. (2020). Random fields simplified: Preserving marginal distributions, correlations, and intermittency, with applications from rainfall to humidity. *Water Resources Research*, 56(2), Article e2019WR026331.
- Papalexiou, S. M., Serinaldi, F., & Porcu, E. (2021). Advancing space-time simulation of random fields: From storms to cyclones and beyond. *Water Resources Research*, 57(8), Article e2020WR029466.
- Porcu, E., Furrer, R., & Nychka, D. (2021). 30 years of space–time covariance functions. *WIREs Computational Statistics*, 13(2), Article e1512. <http://dx.doi.org/10.1002/wics.1512>.
- R Core Team (2023). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing, URL <https://www.R-project.org/>.
- Radojčić, U., & Nordhausen, K. (2024). Order determination in second-order source separation models using data augmentation. In J. Ansari, S. Fuchs, W. Trutschnig, M. A. Lubiano, M. A. Gil, P. Grzegorzewski, & O. Hryniewicz (Eds.), *Combining, Modelling and Analyzing Imprecision, Randomness and Dependence* (pp. 371–379). Cham: Springer.
- Salvana, M. L. O., & Genton, M. G. (2020). Nonstationary cross-covariance functions for multivariate spatio-temporal random fields. *Spatial Statistics*, 37, Article 100411.
- Sigrist, F., Künsch, H. R., & Stahel, W. A. (2012). A dynamic nonstationary spatio-temporal model for short term prediction of precipitation. *The Annals of Applied Statistics*, 6(4), 1452–1477. <http://dx.doi.org/10.1214/12-AOAS564>.
- Sipilä, M., Muehlmann, C., Nordhausen, K., & Taskinen, S. (2024). Robust second-order stationary spatial blind source separation using generalized sign matrices. *Spatial Statistics*, 59, Article 100803. <http://dx.doi.org/10.1016/j.spasta.2023.100803>.
- Sipilä, M., Nordhausen, K., & Taskinen, S. (2024). Nonlinear blind source separation exploiting spatial nonstationarity. *Information Sciences*, 665, Article 120365. <http://dx.doi.org/10.1016/j.ins.2024.120365>.
- Virta, J., & Nordhausen, K. (2017). Blind source separation of tensor-valued time series. *Signal Processing*, 141, 204–216. <http://dx.doi.org/10.1016/j.sigpro.2017.06.008>.
- Virta, J., & Nordhausen, K. (2021). Determining the Signal Dimension in Second Order Source Separation. *Statistica Sinica*, 31, 135–156.
- Wendland, H. (1995). Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics*, 4, 389–396.
- Xu, H., & Gardoni, P. (2018). Improved latent space approach for modelling non-stationary spatial–temporal random fields. *Spatial Statistics*, 23, 160–181.
- Xu, R., Vaida, F., & Harrington, D. P. (2009). Using profile likelihood for semiparametric model selection with application to proportional hazards mixed models. *Statistica Sinica*, 19(2), 819.
- Yan, Y., Huang, H.-C., & Genton, M. G. (2021). Vector autoregressive models with spatially structured coefficients for time series on a spatial grid. *Journal of Agricultural, Biological and Environmental Statistics*, 26(3), 387–408.
- Yi, M., & Nordhausen, K. (2023). Lasso for estimating the signal dimension in ICA. In *2023 31st European Signal Processing Conference (EUSIPCO)* (pp. 2023–2027). <http://dx.doi.org/10.23919/EUSIPCO58844.2023.10289956>.