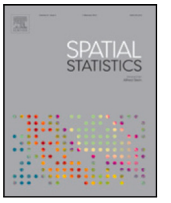
Spatial Statistics 73 (2026) 100971

Contents lists available at ScienceDirect

Spatial Statistics

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



Multivariate low-rank state–space model with SPDE approach for high-dimensional data

Jacopo Rodeschini [a,*], Lorenzo Tedesco [b], Francesco Finazzi [b], Philipp Otto [c], Alessandro Fassò [b]

[a] Department of Engineering and Applied Sciences, University of Bergamo, Dalmine, Italy
[b] Department of Economics, University of Bergamo, Bergamo, Italy
[c] School of Mathematics and Statistics, University of Glasgow, Glasgow, UK

ARTICLE INFO

Keywords:
Spatio-temporal modelling
Stochastic partial differential equations
Gaussian Markov random field
High-dimensional data
Missing data
Sparse matrices
Approximation methods
Italian air quality

ABSTRACT

This paper proposes a novel low-rank approximation of the State–Space Model (SSM) with spatially correlated innovations for the analysis of multivariate spatio-temporal data. The SSM's measurement equation is based on a linear coregionalisation model, which describes the cross-correlation between the observed variables, while the Matérn Gaussian innovation term in the state equation is modelled using the Stochastic Partial Differential Equation (SPDE) approach, allowing a finite-dimensional representation of the latent processes using basis functions defined on spatial meshes. Dimensionality reduction is achieved by appropriately reducing the number of nodes in the meshes. Inference on the model parameters is performed via Maximum Likelihood Estimation (MLE), implemented through the Expectation–Maximisation (EM) algorithm, which features closed-form updating formulas for most parameters and efficient numerical routines for the remainder. We derive theoretical results on the accuracy and convergence of the low-rank approximation and validate them through simulation studies. The EM algorithm and the likelihood derivatives required for inference are implemented in Python/JAX, enabling automatic differentiation and scalable execution across all available local CPU cores, with native support for GPU and TPU acceleration. By analysing a large bivariate air-quality dataset, we demonstrate that reducing the number of nodes by 75% enables model estimation to be 15.8 times faster with only a 15% increase in validation error. We also compare our approach with SPDE-INLA alternatives, demonstrating improved computational scalability while maintaining comparable predictive performance.

1. Introduction

Spatio-temporal statistics is an area of statistical methodology concerned with analysing data that vary across both space and time (Cressie and Wikle, 2011). It has become increasingly important in the geosciences due to extensive sampling campaigns, growing data availability, and its wide range of applications in environmental and climate science.

In many applications, several variables are observed simultaneously. These observations constitute multivariate spatio-temporal data and typically exhibit three types of dependence. Temporal dependence arises from the dynamic nature of underlying processes, whereby past observations influence future states. Spatial dependence, on the other hand, is generally symmetric and instantaneous;

* Corresponding author.
E-mail address: jacopo.rodeschini@unibg.it (J. Rodeschini).

https://doi.org/10.1016/j.spasta.2026.100971
Received 29 August 2025; Received in revised form 19 January 2026; Accepted 11 March 2026
Available online 12 March 2026

nearby locations tend to be similar, while similarity usually diminishes with distance. A third layer of complexity arises from cross-variable dependence, in which correlations exist between variables, either at the same location or across distinct spatial sites.

The most informative scenario is the isotopic configuration, in which all variables are measured at every site, allowing for direct comparison and joint analysis across variables. In contrast, the completely heterotopic configuration refers to the case where each variable is observed only at distinct, non-overlapping locations—that is, not all variables are measured at the same site. In this heterotopic case, spatial relationships among variables must be inferred indirectly because there are no shared sampling points. A less restricted case is the partially heterotopic setting, in which only some variables share some sites. Formal definitions of isotopy and heterotopy are found in Wackernagel (2003).

Modelling both spatial and temporal dependencies is essential for accurate prediction and interpretation (Zhang, 2007), and multivariate spatio-temporal models for heterotopic data have been developed to capture these intricate interdependencies (Finazzi et al., 2013). Such models enhance our ability to uncover underlying processes, improve predictive accuracy, and support more effective policy-making and management decisions.

Among the various modelling approaches, Gaussian Fields (GFs) have become a widely used and powerful tool for solving regression problems in spatial and spatio-temporal data (Rasmussen, 2003). Their popularity stems from their flexibility and their natural ability to capture spatial and temporal dependencies (Stein, 1999). The underlying regression function is assumed to be a realisation of a GF, whose mean and covariance function must be evaluated. Then, using the conditional multivariate normal distribution, a GF provides a closed-form prediction and a predictive interval.

With the widespread availability of remote sensing platforms, especially satellite-based systems, the volume and resolution of spatial data have increased dramatically (see, for instance, Datta et al. 2016, Heaton et al. 2019), thereby significantly amplifying the computational demands associated with GF modelling. This challenge is known as the "Big n problem" (Jona Lasinio et al., 2013): the computational cost of inverting a covariance matrix scales poorly with the number of spatio-temporal observations $n = mT$, where m is the total number of locations and T is the number of time steps, since covariance matrices grow with complexity $\mathcal{O}(m^2T^2)$ in storage and $\mathcal{O}(m^3T^3)$ in inversion. As data volume increases, these demands quickly exceed available resources, rendering exact inference infeasible even with a moderately large n.

In the context of GFs, a range of strategies has been developed to address the high computational cost associated with large datasets. Comprehensive reviews of these methods can be found in Heaton et al. (2019) and Liu et al. (2020). Three major classes of approximation techniques are commonly used: likelihood approximations, low-rank approximations, and sparse representations. A key drawback of using computationally efficient approximated models is the need to assess how closely the approximate predictors and their mean squared prediction errors match the true, theoretical values.

Likelihood approximations, such as the Vecchia approximation (Vecchia, 1988; Katzfuss et al., 2020; Katzfuss and Guinness, 2021), reformulate the joint distribution of observations into a product of conditional distributions, where each observation conditions only on a subset of others. This reduction improves computational efficiency, but the choice of conditioning sets is critical.

Covariance tapering is another approach for handling large spatial datasets, applicable to both univariate (Furrer et al., 2006; Kaufman et al., 2008) and multivariate (Bevilacqua et al., 2016) processes. In tapering, the original covariance function is multiplied by a compactly supported taper function—a function that smoothly goes to zero beyond a certain distance. This operation forces the covariances between sufficiently distant locations to be exactly zero, producing a sparse covariance matrix that greatly reduces computational cost while maintaining much of the original spatial structure. However, tapering assumes that spatial dependence vanishes beyond a fixed distance, which may underestimate long-range correlations (Stein, 1999; Zhang, 2004).

Low-rank approximations reduce dimensionality by expressing the GF as a linear combination of a small number $R \ll m$ of basis functions. Lowering the rank of the covariance matrix significantly cuts computational cost. Well-known example include Fixed-Rank Kriging (Cressie and Johannesson, 2008; Zammit-Mangion and Cressie, 2021) and LatticeKrig (Nychka et al., 2015). However, the choice of basis functions involves multiple tuning parameters, such as the types of basis functions, levels of resolution, and information in the coarsest resolution, which can lead to issues with identifiability and overfitting (Song et al., 2024).

A distinct and powerful class of approximations of spatial GFs is based on Gaussian Markov Random Fields (GMRFs) (Rue and Held, 2005), where conditional independence across a graph structure yields a sparse precision matrix. Lindgren et al. (2011) established a link between Matérn GFs and GMRFs by solving a Stochastic Partial Differential Equation (SPDE) using the Finite Element Method (FEM). This approach enables linear-cost computation of the precision matrix (inverse of the covariance matrix) over complex domains.

The SPDE framework has been significantly extended to spatio-temporal processes. For instance, Clarotto et al. (2024) and Lindgren et al. (2024) explicitly incorporate physical transport and diffusion into the SPDE governing the model.

Regarding the multivariate setting, Bolin and Wallin (2020) generalises the SPDE approach by constructing multivariate GMRFs directly to handle multivariate fields. An alternative, widely used strategy is the Linear Coregionalisation Model (LCM). This framework constructs multivariate dependence by taking a linear combination of independent latent components, each of which satisfies a univariate SPDE. Recent applications of the LCM structure include Dutta et al. (2024), who apply it to positive-valued time series using Bayesian R-INLA framework. Similarly, Gaedke-Merzhäuser et al. (2025) extends the LCM to work with the diffusion specification of the SPDE, achieving scalability by accelerating the full-rank solver on High-Performance Computing (HPC) platforms (GPUs and distributed memory).

The State–Space Model (SSM) is a widely used framework for formulating Gaussian regression problems (Shumway et al., 2000), particularly in spatio-temporal contexts. It explicitly models one-directional temporal dependence through a latent process. Given fixed parameters, inference on the unobserved latent states can be efficiently performed using the Kalman filter, leveraging the

separability of the covariance structure (Hamilton, 2020). In the more common case where model parameters are unknown, the Expectation–Maximisation (EM) algorithm is typically employed for parameter estimation (Shumway et al., 2000). Due to these advantages, SSM have been extensively used in spatio-temporal modelling; see, for instance, Xu and Wikle (2007), Cressie and Wikle (2002), Fasso and Finazzi (2011) and Calculli et al. (2015) for extensions to heterotopic data. The application of SPDE to SSM has been the subject of several previous studies, such as Cameletti et al. (2013), although their formulation is univariate, or Ruiz-Cárdenas et al. (2012). These works, however, fall within the framework of Bayesian approaches and rely on Integrated Nested Laplace Approximations (INLA) (Rue et al., 2009) for estimation.

While inference using the Kalman filter scales linearly with time, $\mathcal{O}(T)$, it incurs a cubic cost in the number of spatial locations, $\mathcal{O}(m^3)$, which becomes prohibitive for large m. To address this issue, several approximation strategies have been proposed. For instance, covariance tapering has been applied to SSMs to induce sparsity in the spatial structure (Bevilacqua et al., 2010). Alternatively, low-rank approximations, such as those based on FRK, have been applied within the SSM framework (Cressie et al., 2010). More recently, Schmidt et al. (2023) proposed a method that reduces rank via a truncated singular value decomposition of the conditional covariance matrix. However, this may come at the cost of losing physical interpretability of the latent process.

In this paper, we propose a novel unified framework that combines state-of-the-art computational strategies. We introduce a multivariate Low-Rank SSM with measurement equation based on a LCM able to describe the cross-correlation between the observed variables, while the Matérn Gaussian innovation term in the state equation is modelled using SPDE-GMRF approach. The temporal evolution statistically via an autoregressive process, avoiding the complexity of physics-driven advection. The model parameters are estimated by maximum likelihood using an EM algorithm combined with a Kalman filter in a frequentist framework. This approach provides closed-form updates for most parameters and naturally accommodates missing data and heterotopic observations. While Bayesian approaches naturally provide parameter uncertainty but may suffer from bias due to prior misspecification, our frequentist framework avoids this issue and leverages automatic differentiation via Python-based JAX library (Bradbury et al., 2018) to efficiently compute likelihood derivatives and standard errors without manual differentiation, while scaling across CPUs and accelerators.

However, we address high-dimensionality not through hardware acceleration, which remains a complementary strategy, but by introducing a low-rank approximation of the latent space estimated via the EM algorithm, allowing for efficient computation on standard resources. This is achieved by leveraging the SPDE framework to significantly reduce the rank of the spatial covariance matrix, from m to $R \ll m$. This approach reduces the computational cost of the Kalman filter to $\mathcal{O}(mR^2T)$, while preserving the physical interpretation of the original model. We prove theoretical results, including asymptotic properties and error bounds for the low-rank model.

The rest of this article is organised as follows. Section 2 introduces the formulation of the Gaussian regression problem using the SSM framework and presents the equations defining a Low-Rank SSM (LR-SSM), tailored for large-scale multivariate spatio-temporal datasets. We also provide theoretical results and examine how the low-rank approximation affects inference. In Section 3, we detail the ML estimation procedure, with a focus on the low-rank representation derived via the SPDE approach. Section 4 presents numerical studies that illustrate the performance of our method and confirm the theoretical results. In Section 5, we present an application of the LR-SSM to a bivariate air quality dataset. Section 6 compares the LR-SSM with Bayesian SPDE-INLA alternatives. Finally, Section 7 concludes the paper with a discussion of the LR-SSM and concluding remarks.

2. The multivariate low-rank state–space model

In this section, we present a multivariate SSM framework and some of its properties. Let $\boldsymbol{y}(\boldsymbol{s},t) = (y_1(\boldsymbol{s},t),\dots,y_p(\boldsymbol{s},t))' \in \mathbb{R}^p$ be a p-dimensional spatio-temporal random field defined over a spatial domain $\Omega \subset \mathbb{R}^2$ and discrete time steps $t = 1,\dots,T$, where $\boldsymbol{s} \in \Omega$ indexes the spatial location. We consider the classical spatio-temporal random effects model (Zammit-Mangion and Cressie, 2021), expressing $\boldsymbol{y}(\boldsymbol{s},t)$ through the measurement equation:

$$\boldsymbol{y}(\boldsymbol{s},t) = \boldsymbol{X}'(\boldsymbol{s},t)\boldsymbol{\beta} + \boldsymbol{W}\boldsymbol{z}(\boldsymbol{s},t) + \boldsymbol{\epsilon}(\boldsymbol{s},t), \tag{1}$$

where $\boldsymbol{X}(\boldsymbol{s},t) = \text{blockdiag}(\boldsymbol{x}_1(\boldsymbol{s},t),\dots,\boldsymbol{x}_p(\boldsymbol{s},t))$ is a $b \times p$ design matrix, $\text{blockdiag}(\cdot)$ denotes the block diagonal matrix operator, $\boldsymbol{x}_i(\boldsymbol{s},t)$ is a column vector of dimension b_i associated with $y_i(\boldsymbol{s},t)$ for $i = 1,\dots p$, and $b = \sum_{i=1}^{p} b_i$. The parameter $\boldsymbol{\beta} = (\boldsymbol{\beta}_1',\dots,\boldsymbol{\beta}_p')'$ is a b-dimensional vector of regression coefficients, where each $\boldsymbol{\beta}_i$, for $i = 1,\dots,p$, is a column vector of dimension b_i corresponding to the regression covariates associated with $y_i(\boldsymbol{s},t)$. $\boldsymbol{W}$ is a $p \times q$ loading matrix that provides a linear mapping from the q-dimensional latent random effects $\boldsymbol{z}(\boldsymbol{s},t)$ to the p-dimensional observation space. In particular, each row of $\boldsymbol{W}$ specifies how the components of $\boldsymbol{z}(\boldsymbol{s},t)$ combine to form each observed variable. The term $\boldsymbol{\epsilon}(\boldsymbol{s},t)$ represents the Gaussian measurement error, assumed to be statistically independent across components, space and time. We denote by $\boldsymbol{\Sigma} = \text{diag}(\boldsymbol{\sigma}^2)$ the diagonal variance matrix of $\boldsymbol{\epsilon}(\boldsymbol{s},t)$, with $\boldsymbol{\sigma}^2 = (\sigma_1^2,\dots,\sigma_p^2)' \in \mathbb{R}_+^p$ being the vector of measurement error variances.

The latent process $\boldsymbol{z}(\boldsymbol{s},t)$ evolves over time, following the state equation:

$$\boldsymbol{z}(\boldsymbol{s},t) = \boldsymbol{F}\boldsymbol{z}(\boldsymbol{s},t-1) + \boldsymbol{\eta}(\boldsymbol{s},t), \tag{2}$$

where $\boldsymbol{F} = \text{diag}(\boldsymbol{f})$ denotes a $q \times q$ diagonal transition matrix describing the system's dynamics, and $\boldsymbol{f} = (f_1,\dots,f_q)'$ with each $f_i \in (-1,1)$ to ensure the stationarity of $\boldsymbol{z}(\boldsymbol{s},t)$. For every $\boldsymbol{s} \in \Omega$, $\boldsymbol{z}(\boldsymbol{s},0)$ is an independent Gaussian variable with mean $\boldsymbol{\mu}_0 \in \mathbb{R}^q$ and $q \times q$ covariance matrix $\boldsymbol{\Sigma}_0$. The term $\boldsymbol{\eta}(\boldsymbol{s},t) = (\eta_1(\boldsymbol{s},t),\dots,\eta_q(\boldsymbol{s},t))'$ represents a q-dimensional, zero-mean Gaussian innovation process which is temporally independent. Moreover, we assume that the components of $\boldsymbol{\eta}(\boldsymbol{s},t)$ are mutually independent, i.e., $\eta_j(\boldsymbol{s},t) \perp\!\!\!\perp \eta_{j'}(\boldsymbol{s}',t')$ for all $j \neq j'$, at any spatial locations $\boldsymbol{s},\boldsymbol{s}'$ and times t,t'. We also assume that each $\eta_i(\boldsymbol{s},t)$ is a zero-mean Màtern

GF, with scale parameter κ_i, unit marginal variance and smoothness parameter ν_i. Specifically, the Matérn covariance function between two spatial locations $\boldsymbol{s}$ and $\boldsymbol{s}'$ is given by

$$\mathrm{Cov}(\eta_i(\boldsymbol{s},t),\eta_i(\boldsymbol{s}',t)) = \frac{1}{\Gamma(\nu_i)2^{\nu_i-1}}(\kappa_i\|\boldsymbol{s}-\boldsymbol{s}'\|)^{\nu_i}K_{\nu_i}(\kappa_i\|\boldsymbol{s}-\boldsymbol{s}'\|), \tag{3}$$

where K_{ν_i} is the modified Bessel function of the second kind, and κ_i controls the spatial range of dependence. The smoothness parameter is rarely estimated directly; instead, it is typically set a priori based on the desired level of smoothness in the Matérn process or informed by physical assumptions about the underlying phenomenon.

Together, the measurement and state Eqs. (1)–(2) define the linear SSM (Shumway et al., 2000). This formulation is particularly appealing due to its generality and flexibility, and can capture both observed and latent components of the time series. The model parameter set that needs to be estimated is:

$$\Pi = \{\boldsymbol{\beta},\boldsymbol{\sigma}^2,\boldsymbol{f},\boldsymbol{W},k_1,\dots,k_q\}. \tag{4}$$

Inference on $\boldsymbol{z}(\boldsymbol{s},t)$ is carried out via the Kalman filter, which entails inversion of an $m\times m$ matrix at each time step $t=1,\dots,T$, leading to a computational complexity of $\mathcal{O}(m^3T)$. While linear in time, this complexity becomes prohibitive for large m, thereby motivating the low-rank approximation described in the next section

2.1. The low-rank SPDE approximation

In this section, we introduce a low-rank SSM, based on the SPDE approach (Lindgren et al., 2011), designed to approximate the data-generating process (1)–(2) and to enable inference on the latent dynamics at a reduced computational cost of $\mathcal{O}(mR^2T)$ with $R \ll m$. To this end, we discuss how to replace the GF η_i in Eq. (2) with a low-rank approximation by expressing η_i as a linear combination of R finite basis functions. We also examine the special case $R=m$, which is of interest because, from a theoretical standpoint, the SPDE approach always provides a finite-dimensional approximation to an inherently infinite-dimensional Gaussian field. Consequently, even when $R=m$, the representation is low-rank relative to the true complexity of the latent spatial process.

Whittle (1954, 1963) showed that a Matérn GF can be characterised as the unique solution to a specific stochastic partial differential equation (SPDE) defined over $\mathbb{R}^d$. Then, restricting the domain to Ω, and applying the Galerkin method (Ern and Guermond, 2004) to this SPDE, one obtains a low-rank approximation of the field η_i, which corresponds to a Gaussian Markov Random Field (GMRF) representation of the original GF. The theoretical foundation for the proposed approximation is the explicit connection between the GMRF and the Matérn GF given by Lindgren et al. (2011). To simplify the presentation, we present the resulting low-rank representation in this section and defer the technical details to Appendix C.

Let us consider a partition of the domain $\Omega\subset\mathbb{R}^2$ into a set of non-overlapping triangles, i.e., $\Omega=\bigcup_i E_i$, where any two triangles share at most a common edge or vertex. The result of the discretisation is a mesh $\mathcal{G}_R=(\mathcal{V}_R,\mathcal{E}_R)$, with R equal to the cardinality of the vertices $\mathcal{V}_R=\{\boldsymbol{r}_1,\dots,\boldsymbol{r}_R\}$ and $\mathcal{E}_R$ being the set of the edges of the triangles. Based on the mesh $\mathcal{G}_R$, we define a set of basis functions $\boldsymbol{\psi}_R(\boldsymbol{s})=(\psi_1(\boldsymbol{s}),\dots,\psi_R(\boldsymbol{s}))'$, where each $\psi_i:\mathbb{R}^2\to\mathbb{R}$ is selected from the class of piecewise linear functions with compact support. These functions are constructed such that $\psi_i(\boldsymbol{r}_k)=1$ if $i=k$, and $\psi_i(\boldsymbol{r}_k)=0$ if $i\neq k$, for all $i,k=1,\dots,R$. Additionally, we assume that the process $\eta_i(\boldsymbol{s},t)$ satisfies a Neumann boundary condition on $\partial\Omega$,imposed weakly through the variational formulation in the finite element approximation. This ensures that the SPDE representation inherits the desired asymptotic properties, as discussed in the sequel. A similar treatment is used in Lindgren et al. (2011).

Let $\boldsymbol{\eta}_j(\mathcal{V}_R,t)$ denote the vector:

$$\boldsymbol{\eta}_j(\mathcal{V}_R,t)=\big(\eta_j(\boldsymbol{r}_1,t),\dots,\eta_j(\boldsymbol{r}_R,t)\big)',$$

for $j=1,\dots,q$, where $\mathcal{V}_R=\{\boldsymbol{r}_1,\dots,\boldsymbol{r}_R\}$ is a finite set of spatial locations. Then, we define the full vector $\boldsymbol{\eta}(\mathcal{V}_R,t)\in\mathbb{R}^{qR}$ as:

$$\boldsymbol{\eta}(\mathcal{V}_R,t)=\big(\boldsymbol{\eta}_1(\mathcal{V}_R,t)',\dots,\boldsymbol{\eta}_q(\mathcal{V}_R,t)'\big)'.$$

The extension to the case where each component has a different mesh, i.e. $\mathcal{G}_R^j=(\mathcal{V}_R^j,\mathcal{E}_R^j)$ for $j=1,\dots,q$, is straightforward. Here, for simplicity and without loss of generality, we assume that the latent process components share the same finite element mesh $\mathcal{G}_R$. In Section 5, we provide an example with disjoint meshes. Similarly, let $\boldsymbol{z}_j(\mathcal{V}_R,t)$ for $j=1,\dots,q$ be the vector $\boldsymbol{z}_j(\mathcal{V}_R,t)=(z_j(\boldsymbol{r}_1,t),\dots,z_j(\boldsymbol{r}_R,t))'$ and $\boldsymbol{z}(\mathcal{V}_R,t)$ be the vector $\boldsymbol{z}(\mathcal{V}_R,t)=(\boldsymbol{z}_1(\mathcal{V}_R,t)',\dots,\boldsymbol{z}_q(\mathcal{V}_R,t)')'$.

Let $\mathcal{H}^1(\Omega)$ denote the first-order Sobolev space over the spatial domain Ω. Consider the finite-dimensional subspace $\mathcal{H}^1_R(\Omega)\subset\mathcal{H}^1(\Omega)$ spanned by basis functions $\{\psi_i\}_{i=1}^R$. We approximate each component of the innovation term in (2) using the Galerkin solution of the SPDE defining the Matérn GF (see Appendix C, Eq. (12)), projected onto $\mathcal{H}^1_R(\Omega)$. This yields the following approximation:

$$\boldsymbol{z}(\boldsymbol{s},t)\approx\boldsymbol{F}\boldsymbol{z}(\boldsymbol{s},t-1)+\boldsymbol{\Psi}_R(\boldsymbol{s})\boldsymbol{\eta}(\mathcal{V}_R,t), \tag{5}$$

where $\boldsymbol{\Psi}_R(\boldsymbol{s})=\boldsymbol{I}_p\otimes\boldsymbol{\psi}_R(\boldsymbol{s})'$, $\boldsymbol{\psi}_R(\boldsymbol{s})=(\psi_1(\boldsymbol{s}),\dots,\psi_R(\boldsymbol{s}))'$ and $\otimes$ denotes the Kronecker product.

We propose to leverage (5) to obtain a low-rank approximation of model (1)–(2). That is, considering (5) holding as equality, we replace (1)–(2) with the following LR-SSM:

$$\boldsymbol{y}(\boldsymbol{s},t)=\boldsymbol{X}'(\boldsymbol{s},t)\boldsymbol{\beta}+\boldsymbol{W}\boldsymbol{\Psi}_R(\boldsymbol{s})\boldsymbol{z}(\mathcal{V}_R,t)'+\boldsymbol{\varepsilon}(\boldsymbol{s},t), \tag{6}$$

$$\boldsymbol{z}(\mathcal{V}_R,t)=(\boldsymbol{F}\otimes\boldsymbol{I}_R)\boldsymbol{z}(\mathcal{V}_R,t-1)+\boldsymbol{\eta}(\mathcal{V}_R,t). \tag{7}$$

In Eq. (7), the continuous latent field $\boldsymbol{z}(\boldsymbol{s},t)$ from (1) is replaced by a low-rank approximation of the form $\boldsymbol{\Psi}_R(\boldsymbol{s})\boldsymbol{z}(\mathcal{V}_R,t)'$, where the reduction ratio is defined as $LR = \frac{|\mathcal{V}_R|}{m} \times 100\%$. Comparing (1) with (6), the approximation introduces a model-based error. In Section 2.2, we show that the model-based error can be controlled by the number of vertices R. As expected, a finer finite element mesh yields higher inference accuracy and ensures that the mesh serves purely as an approximation device rather than a modelling assumption.

We summarise here the advantages of using the model defined in Eqs. (6)–(7) to make inference on model (1)–(2). First, the low-rank approximation reduces the computational complexity of the Kalman filter from $\mathcal{O}(m^3T)$ to $\mathcal{O}(mR^2T)$, gaining efficiency for $R \ll m$. The practitioner can therefore control the balance between computational cost and accuracy by choosing the value of R. Second, it preserves the first- and second-order moments of the original process in the limit, thereby enabling scalable inference and prediction in high-dimensional spatio-temporal settings. Third, once the random coefficients $\boldsymbol{z}(\mathcal{V}_R,t)$ are estimated, evaluating the latent process at new locations has a linear computational cost. This enables efficient construction of spatial maps of the observed processes, making the LR-SSM competitive even when $R = m$ ($LR = 100\%$). Details on model estimation and spatial mapping are provided in Section 3. Most model parameters can be estimated via closed-form expressions. For the rescaling parameters, k_i for $i = 1,\dots,q$, although no closed-form solution is available, the assumption of independent innovation components allows for efficient minimisation whose complexity grows linearly with the dimension of the latent component q. This makes the estimation procedure particularly efficient as compared to other SSMs such as those in Calculli et al. (2015).

2.2. Identifiability, convergence, and approximation bounds

We begin by analysing the theoretical properties of the proposed approximation. For clarity, we first study the LR-SSM (6)–(7) for a fixed mesh, which provides a finite-dimensional representation of the model. This allows us to establish point identification in a distributional sense. We then investigate how the LR-SSM behaves as the mesh is refined: we show that the low-rank approximation of the latent process converges weakly to the continuous Matérn field and derive a uniform bound on the discrepancy between the outputs of the LR-SSM and those of the full continuous SSM. Together, these results quantify the approximation error and establish that it vanishes as the mesh resolution increases.

Assumption 1. For a subset $S \subset \Omega$, assume:

(i) The limits

$$\boldsymbol{M}_X = \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T}\int_S \boldsymbol{X}(\boldsymbol{s},t)\,d\boldsymbol{s}, \qquad \boldsymbol{L}_X = \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T}\int_S \boldsymbol{X}(\boldsymbol{s},t)\boldsymbol{X}(\boldsymbol{s},t)'\,d\boldsymbol{s},$$

exist and are finite.

(ii) $\boldsymbol{L}_X$ is nonsingular.

(iii) $\boldsymbol{X}(\boldsymbol{s},t)$ is non-random or, more generally,

$$\mathbb{E}\left[\boldsymbol{X}(\boldsymbol{s},t)\,\boldsymbol{W}\boldsymbol{\Psi}_R(\boldsymbol{s})\boldsymbol{z}_t\right] = \boldsymbol{0}, \qquad \mathbb{E}[\boldsymbol{X}(\boldsymbol{s},t)\epsilon(\boldsymbol{s},t)] = \boldsymbol{0},$$

and $\sup_{(\boldsymbol{s},t)} \|\boldsymbol{X}(\boldsymbol{s},t)\| < \infty$.

Assumption 1 imposes a standard conditions on the fixed-effects component of the model, ensuring identifiability of the regression coefficient vector $\boldsymbol{\beta}$.

Assumption 2. (*i*) $f_1 > f_2 > \cdots > f_q$; (*ii*) $\boldsymbol{W}$ has full column rank, with each column scaled so that its first non-zero element is positive; (*iii*) $\boldsymbol{\Psi}_R(\boldsymbol{s}) = \boldsymbol{I}_p \otimes \boldsymbol{\psi}_R(\boldsymbol{s})'$ is known and has full row rank; (*iv*) The map $k \mapsto \boldsymbol{Q}_k$ is injective.

Assumption 2 imposes restrictions on the parameter space, excluding alternative representations of the LR-SSM that may result from permuting the order or changing the sign of latent components. Condition (iv) is easy to be verified when $\nu_i = 1$ for all $i = 1,\dots,q$, as we show in Lemma 1 included in Appendix B.

We consider point identification in a distributional sense, where model parameters are uniquely determined by the joint distribution of the observables (Lewbel, 2019), meaning that no two distinct parameter sets Π and $\Pi^\star$ can generate the same distribution of the observed process.

Theorem 1 (*Identifiability*)**.** *Let Assumptions 1–2 hold, and consider the data generated according to the LR-SSM (6)–(7), with parameter set Π. Assume that the process $\{\boldsymbol{y}(\boldsymbol{s},t) : (\boldsymbol{s},t) \in S \times \mathbb{N}\}$ is observed over a spatially dense set $S \subset \Omega$ and over an arbitrarily long time horizon. Then, the parameter set Π is identifiable*

The proof provided in Appendix B.

The SSM described in (1)–(2) provides a continuous representation of the observed random process, capturing its inherent continuity. We now analyse the estimation errors that arise when the LR-SSM is employed to infer data generated by the true SSM.

First, we discuss the convergence of the low-rank approximation of the latent component. Let x^R be a sequence of $L^2(\Omega)$-bounded Gaussian fields. As defined in Lindgren et al. (2011), we say $x^R \xrightarrow{D\{L^2(\Omega)\}} x$ if for all $f, g \in L^2(\Omega)$,

$$\mathbb{E}[\langle f, x^R\rangle] \to \mathbb{E}[\langle f, x\rangle], \quad \text{and} \quad \mathrm{Cov}(\langle f, x^R\rangle, \langle g, x^R\rangle) \to \mathrm{Cov}(\langle f, x\rangle, \langle g, x\rangle).$$

As $\mathcal{H}^1_R(\Omega)$ is dense in $\mathcal{H}^1(\Omega)$ as $R \to \infty$, Theorem 3 of Lindgren et al. (2011) ensures that for a fixed $t \in \{1, \dots, T\}$, it holds $\boldsymbol{\Psi}(\boldsymbol{s})\boldsymbol{\eta}(\mathcal{V}_R, t) \xrightarrow{D\{L^2(\Omega^q)\}} \boldsymbol{\eta}(\boldsymbol{s}, t)$. Thanks to the Markovian structure of $\boldsymbol{z}(\boldsymbol{s}, t)$, the preceding result ensures that, for any finite time horizon T, the approximate latent field converges to the true latent field. This conclusion is stated precisely in the following corollary of Theorem 5 (Lindgren et al., 2011), as convergence of each component implies convergence of the finite-dimensional collection.

Corollary 1 (*Weak Convergence*). *For every $t = 1, \dots, T$ and $i = 1, \dots, q$, let $\eta_i(\boldsymbol{s}, t)$ satisfy the Matérn SPDE (12) with homogeneous Neumann condition, and define $z_i(\boldsymbol{s}, t) = \sum_{r=0}^{t} f_i^{t-r} \eta_i(\boldsymbol{s}, r)$. Let $\boldsymbol{\eta}_i(\mathcal{V}_R, t)$ be the vector of coefficients of the Galerkin solution of the SPDE in $\mathcal{H}^1_R(\Omega, \kappa)$. Then*

$$\boldsymbol{\Psi}_R(\boldsymbol{s})\boldsymbol{z}(\mathcal{V}_R, t) \xrightarrow{D\{L^2(\Omega^{qT})\}} \boldsymbol{z}(\boldsymbol{s}, t),$$

where $\boldsymbol{z}(\mathcal{V}_R, t)$ are defined in terms of $\{\boldsymbol{\eta}_i(\boldsymbol{s}, t)\}_{i=1}^q$ in (7). In particular,

$$\mathbb{E}[\boldsymbol{\Psi}_R(\boldsymbol{s})\boldsymbol{z}(\mathcal{V}_R, t)] \to \mathbb{E}[\boldsymbol{z}(\boldsymbol{s}, t)],$$
$$\mathrm{Cov}(\boldsymbol{\Psi}_R(\boldsymbol{s})\boldsymbol{z}(\mathcal{V}_R, t), \boldsymbol{\Psi}_R(\tilde{\boldsymbol{s}})\boldsymbol{z}(\mathcal{V}_R, \tilde{t})) \to \mathrm{Cov}(\boldsymbol{z}(\boldsymbol{s}, t), \boldsymbol{z}(\tilde{\boldsymbol{s}}, \tilde{t})).$$

The proof is provided in Appendix C.1, and relies on standard assumptions about the finite element mesh topology, namely that the minimal mesh angles are bounded away from zero and that the maximum edge length tends to zero. Notably, the approximation preserves the first- and second-order moments of the original process in the limit, thereby enabling scalable inference and prediction in high-dimensional spatio-temporal settings.

As a second result, we provide a bound on the observed process when the original model (1)–(2) is approximated by the model (6)–(7).

Theorem 2 (*Bound on the Observable Process*). *For every $t = 1, \dots, T$ and $\boldsymbol{s} \in \Omega$, let $\boldsymbol{y}(\boldsymbol{s}, t)$ be a random process satisfying the model (1)–(2) with set of parameter Π. Let $\boldsymbol{y}^R(\boldsymbol{s}, t)$ the discretised version of $\boldsymbol{y}(\boldsymbol{s}, t)$ obtained by projecting each component of $\boldsymbol{\eta}_t$ into $\mathcal{H}_R(\Omega)$, so that $\{\boldsymbol{y}_t^R\}_{i=1}^T$ satisfies model (6)–(7) with the same parameter set Π. The following bound holds:*

$$\mathbb{E}\|\boldsymbol{y}_t - \boldsymbol{y}_t^R\|_2^2 \leq C \, \frac{1 - \varphi^{2(t+1)}}{1 - \varphi^2} \, h^2,$$

where C is a constant that depends on $\|\boldsymbol{W}\|$ and q, $\varphi = \max_i |f_i| < 1$, h denotes the maximum edge length of the finite element mesh used in the approximation, and $\|\cdot\|_2$ stands for the $L^2(\Omega)$–norm.

We prove this result in Appendix C.2.

Theorem 3 shows that the parameter bias when LR-SSM and SSM are parametrised by the same set Π, is uniformly bounded in time since $\frac{1-\varphi^{2(t+1)}}{1-\varphi^2} \leq \frac{1}{1-\varphi^2}$ as $t \to \infty$. Thus, the discrepancy between the true and reduced-order outputs can never grow without bound; at worst, it is proportional to the finite element mesh width h. The linear dependence on h reflects the usual accuracy–cost trade-off: a finer mesh reduces the error while demanding greater computational effort. The constant C depends on $\|\boldsymbol{W}\|$ and q, since the matrix $\boldsymbol{W}$ modulates how the approximation error introduced by replacing $\boldsymbol{z}$ with $\boldsymbol{z}^R$ propagates to the observable output $\boldsymbol{y}$.

Theorem 3 (*Bound on Model Parameters*). *Under Assumptions 1–2, for a fixed R, the bias in the estimation of the vector parameters $\boldsymbol{\pi} = (\boldsymbol{\beta}', \boldsymbol{\sigma}', \mathrm{vec}(\boldsymbol{W})', \boldsymbol{f}', k_1, \dots, k_q)'$ using the LR-SSM (6)–(7) for data generated by the SSM (1)–(2) is of order $\mathcal{O}(h)$, where h denotes the maximum edge length of the finite element mesh used in the approximation. Specifically,*

$$\left\| \boldsymbol{\pi}_R^\star - \boldsymbol{\pi} \right\| = \mathcal{O}(h),$$

where $\boldsymbol{\pi}_R^\star$ is the pseudo-true parameter vector minimising the Kullback–Leibler divergence (D_{KL}) between the true marginal law $\mathcal{P}_{\boldsymbol{\pi}}$ of $\boldsymbol{y}(\boldsymbol{s}, t)$ with parameter vector $\boldsymbol{\pi}$ and the misspecified LR-SSM marginal law $\mathcal{P}_{\boldsymbol{\vartheta}}^{(R)}$ of $\boldsymbol{y}^R(\boldsymbol{s}, t)$ with vector of parameter $\boldsymbol{\vartheta}$, i.e. $\boldsymbol{\pi}_R^\star = \mathrm{argmin}_{\boldsymbol{\vartheta}} D_{KL}(\mathcal{P}_{\boldsymbol{\vartheta}}^{(R)} \| \mathcal{P}_{\boldsymbol{\pi}})$. Furthermore, the parameter $\boldsymbol{\beta}$ has no bias.

The proof is reported in Appendix D.

While we do not provide a formal proof of consistency for the ML estimator under the LR-SSM approximation, Corollary 1, Theorems 2 and 3 offer an intuitive justification for why consistency is plausible. Corollary 1 shows that, as the mesh is refined ($h \to 0$), the LR-SSM reproduces the joint distribution of the true SSM, while Theorem 3 establishes that the pseudo-true parameter, the target of the ML estimator under misspecification, converges to the true parameter at rate $\mathcal{O}(h)$. Taken together, these results suggest that, if the approximation error vanishes and the amount of spatial and/or temporal information grows sufficiently, the ML estimator converges to the true parameter vector. However, rigorously establishing such a consistency result requires addressing the combined challenges of misspecified likelihoods, the state–space structure of the model, and assumptions about the sampling regime; we leave this analysis to future work.

Taken as a whole, Corollary 1, Theorems 2 and 3 provide a comprehensive theoretical foundation supporting the use of the LR-SSM as a principled and efficient approximation to the full SSM. They guarantee that both the latent process and the parameter estimates become increasingly accurate as the mesh is refined, thereby justifying the adoption of the LR-SSM in practical spatio-temporal inference and prediction tasks.

3. Parameter estimation

We estimate the parameter set Π using the maximum likelihood (ML) approach, implemented via the EM algorithm. This method is particularly well-suited for models with latent structures. Our objective is to estimate Π based on the observed data. Specifically, we observe each component $y_j(s,t)$ at spatial locations $S_{j,t} = \{s_{i,t}\}_{i=1}^{m_{j,t}}$.

We assume that the mesh $\mathcal{G}_R$, which defines the spatial domain for the latent process within the state–space model described in Eqs. (6)–(7), is given. The procedure used to construct this mesh is detailed in Section 3.1.

The detailed EM algorithmic steps, the closed-form expressions for the parameter updates, as well as the matrix formulation of the model, are presented in Appendix E. We denote by $\boldsymbol{Q}_{\kappa_i}^{-1}$ the covariance matrix of $\boldsymbol{\eta}_i(\mathcal{V}_R,t)$, as specified in (3). In Section 3.2, we show how the precision matrix $\boldsymbol{Q}_{\kappa_i}$ is constructed at linear cost using the FEM (Lindgren et al., 2011). Thus, the evaluation of the objective function in the maximisation step, which involves computing the log-determinant of $\boldsymbol{Q}_{\kappa_i}$, benefits on the one hand from the sparsity of the matrix and, on the other, from the low-rank approximation that limits its effective dimension, making the overall optimisation step computationally fast.

Note that given the MLE of the parameter set Π, the plug-in predictions at new site $s_0 \in \Omega$ and time $t = 1,\dots,T$ are given by

$$\hat{\boldsymbol{y}}(\boldsymbol{s_0},t) = \boldsymbol{X}'(\boldsymbol{s_0},t)\hat{\boldsymbol{\beta}} + \hat{\boldsymbol{W}}\boldsymbol{\Psi}_R(s_0)\hat{\boldsymbol{z}}^T(\mathcal{V}_R,t), \tag{8}$$

$$\boldsymbol{\Sigma}_{\hat{\boldsymbol{y}}}(\boldsymbol{s_0},t) = \hat{\boldsymbol{W}}\boldsymbol{\Psi}_R(s_0)\,\mathrm{Var}[\hat{\boldsymbol{z}}^T(\mathcal{V}_R,t)]\boldsymbol{\Psi}_R(s_0)'\hat{\boldsymbol{W}}', \tag{9}$$

where $\hat{\boldsymbol{z}}^T(\mathcal{V}_R,t)$ denotes the smoothed state process $\boldsymbol{z}(\mathcal{V}_R,t)$, with associated variance $\mathrm{Var}[\hat{\boldsymbol{z}}^T(\mathcal{V}_R,t)]$, estimated by the Kalman smoother variance $\boldsymbol{P}_t^T$ (see the Kalman smoother derivation in Appendix E). The vectors $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{W}}$ are the MLE of $\boldsymbol{\beta}$ and $\boldsymbol{W}$, respectively. Eq. (8) results in the BLUP procedure, and it relies only on linear algebra operations with a linear computational cost of $\mathcal{O}(R)$.

Fig. 1 offers a visual overview of the main methodological components and summarises the full workflow of LR-SSM.

3.1. Choice of the mesh

Corollary 1 establishes the weak convergence $\boldsymbol{\Psi}_R(s)\boldsymbol{z}(\mathcal{V}_R,t) \to \boldsymbol{z}(s,t)$. This result holds under standard assumptions on the finite element mesh, say $\mathcal{G}_R = (\mathcal{V}_R,\mathcal{E}_R)$, specifically that the minimal mesh angles are bounded away from zero, and the maximum edge length tends to zero.

Since the choice of triangulation significantly affects both the quality of the approximation and the asymptotic convergence, it is essential to construct finite element meshes that satisfy these topological criteria. In this section, we introduce a method for generating such meshes, involving R vertices, where R is chosen to balance computational tractability with accurate and reliable parameter estimation.

Let S denote the set of all observed locations, that is, $S = \bigcup_{i=1}^{p}\bigcup_{t=1}^{T} S_{i,t}$, where $S_{i,t}$ is the set of locations at which variable i is observed at time t.

Among available options, Delaunay triangulations are commonly used to construct finite element meshes because they maximise the smallest angle among all possible triangulations of a given set of points, thereby avoiding the creation of overly narrow triangles. We construct an initial triangulation $\mathcal{G}_m$ by applying the Delaunay algorithm to the full set S, where $m = |S|$. To reduce the mesh to a fixed number R of vertices, we apply an iterative procedure. At each step, the node associated with the smallest triangle area is removed, followed by local re-triangulation to preserve the Delaunay structure. This process is repeated until exactly R vertices remain. The resulting triangulation defines the mesh $\mathcal{G}_R = (\mathcal{V}_R,\mathcal{E}_R)$. To ensure that the interior angles in the mesh are suitably bounded away from zero, we apply standard Laplacian smoothing (Field, 1988) to the triangular finite mesh $\mathcal{G}_R$. Laplacian smoothing is an effective ad hoc method for improving mesh quality, particularly in regions with varying element density, by adjusting vertex positions to produce a more regular triangulation. The Laplacian smoothing algorithm for Delaunay triangulations is presented in pseudocode in Algorithm 1. It produces a Laplacian-smoothed version of the Delaunay triangulation, denoted by $\mathcal{G}_R^* = (\mathcal{V}_R^*,\mathcal{E}_R^*)$. The resulting set of smoothed vertices $\mathcal{V}_R^*$ can then be used in place of $\mathcal{V}_R$ in the state equation of the SSM model (6)–(7). For additional details and visual illustrations demonstrating the effectiveness of this approach in improving angle quality relative to standard Delaunay triangulations, we refer the reader to Field (1988).

Algorithm 1: Laplacian smoothing algorithm for Delaunay triangulations.

Input: Triangulation $\mathcal{G}_R = (\mathcal{V}_R,\mathcal{E}_R)$; angle threshold $\theta_{\min}$
Output: Triangulation with all angles $\geq \theta_{\min}$

```
while any triangle angle < θ_min do
    foreach interior vertex r ∈ V_R do
        Let {r_1, ..., r_k} be the neighbours of r;
        Compute r* ← (1/k) Σ_{i=1}^{k} r_i;
        if r* preserves Delaunay and connectivity then
            Update position: r ← r*;
        else
            Remove r; locally retriangulate;
            Insert vertex at r*;
```

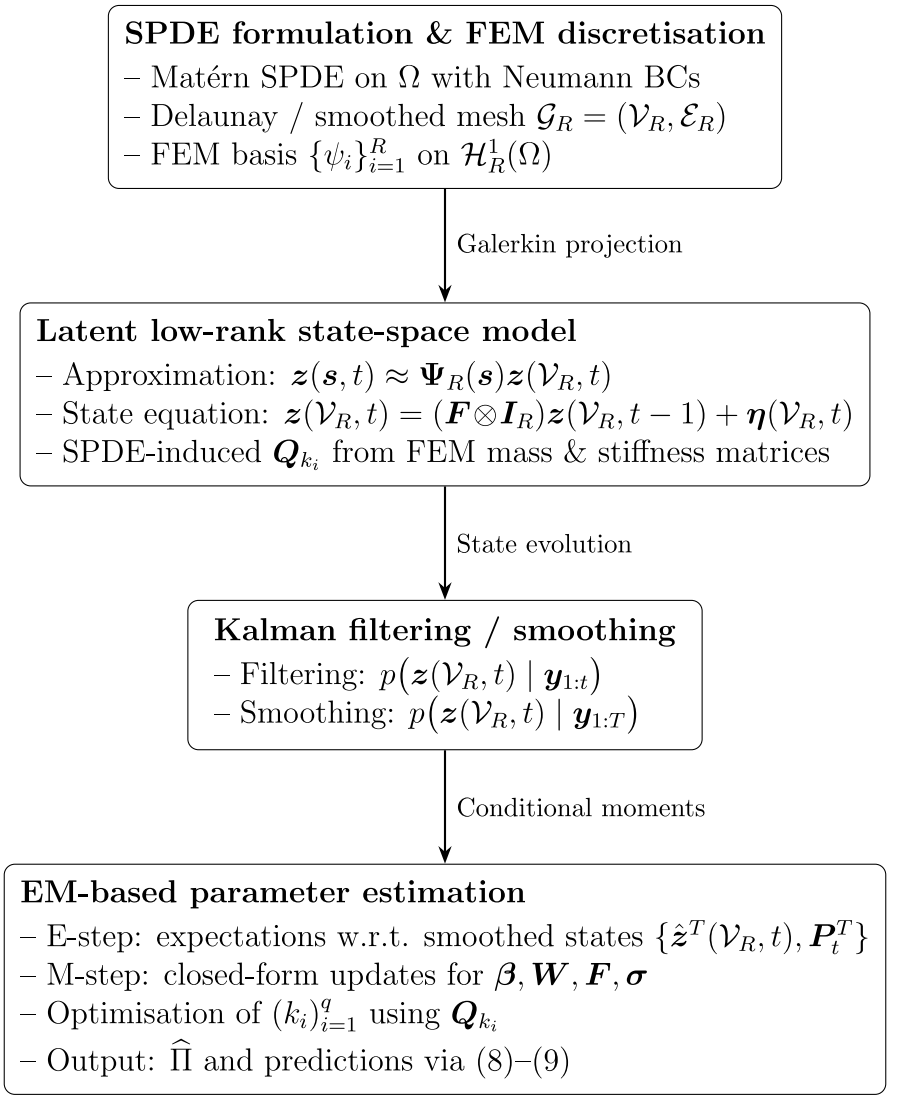


Fig. 1. Schematic workflow of the LR-SSM. The SPDE is discretised via FEM to construct a low-rank latent state–space representation, temporal dependence is handled through Kalman filtering and smoothing, and parameters are estimated via an EM algorithm.

3.2. Approximation of the precision matrix

In this section, we briefly recall how the approximate precision matrix $\boldsymbol{Q}$ is obtained within the SPDE framework of Lindgren et al. (2011); further details are given in Appendix C. In particular, we set $\nu_i = 1$, which corresponds to a spatial process that is continuous but not mean-square differentiable. This assumption is adopted throughout the entire implementation and strikes a balance between model flexibility and computational efficiency, making it well suited for applications where a moderate degree of spatial roughness is expected or acceptable.

Let $\mathcal{G}_R = (\mathcal{V}_R, \mathcal{E}_R)$ be a triangulation of the spatial domain Ω with $R = |\mathcal{V}_R|$ vertices associated with piecewise-linear basis functions $\{\psi_i\}_{i=1}^R$ of the finite element space. Let $\boldsymbol{C}$ and $\boldsymbol{G}$ denote the mass matrix and stiffness matrix, respectively, defined as

$$\boldsymbol{C}_{ij} = \langle \psi_i, \psi_j \rangle, \qquad \boldsymbol{G}_{ij} = \langle \nabla\psi_i, \nabla\psi_j \rangle.$$

We then define the matrix

$$\boldsymbol{K}_{\kappa_i^2} = \kappa_i^2 \boldsymbol{C} + \boldsymbol{G}.$$

Using the Neumann boundary conditions, the finite-dimensional representations of the solutions to Eq. (12) have precision

$$\boldsymbol{Q}_{k_i} = \boldsymbol{K}_{\kappa_i^2}\, \boldsymbol{C}^{-1}\, \boldsymbol{K}_{\kappa^2}. \tag{10}$$

The matrix $\boldsymbol{C}^{-1}$ is dense, which leads to a dense precision matrix in the resulting model. To address this issue, Lindgren et al. (2011, Appendix C.5) show that $\boldsymbol{C}$ can be approximated by a diagonal matrix $\widetilde{\boldsymbol{C}}$, where $\widetilde{C}_{ii} = \langle \psi_i, 1 \rangle$. This approximation yields a sparse precision matrix, resulting in a GMRF representation of the original Matérn GF with scale k_i. Because each basis function overlaps only a fixed, small number of neighbours, $\widetilde{\boldsymbol{C}}$, $\boldsymbol{G}$, $\boldsymbol{K}_{\kappa_i^2}$ contain $\mathcal{O}(R)$ non-zero entries. Thus, all matrix products can be assembled with linear complexity $O(R)$. Due to the independence assumption of the q components of the random vector $\boldsymbol{\eta}(\mathcal{V}_R, t)$, the resulting precision matrix is $\boldsymbol{Q} = \text{blockdiag}(\boldsymbol{Q}_{k_1}, \dots, \boldsymbol{Q}_{k_q})$

The result in (10) is derived using Neumann boundary conditions to solve the SPDE (12). We summarise the main considerations about the boundary effect and the boundary-mitigation strategy in the next paragraph and refer the reader to Lindgren et al. (2011) for further details.

3.2.1. Boundary effects

Imposing Neumann boundary conditions on the SPDE induces a folding effect in the resulting covariance function. This behaviour, analysed in detail by Lindgren et al. (2011), implies that the covariance of the SPDE solution on a bounded domain closely matches the stationary Matérn covariance whenever locations lie at least two Matérn ranges away from the boundary. To exploit this property, we enlarge the computational domain by adding a set of auxiliary vertices outside the region of interest. Specifically, we construct an augmented mesh

$$\mathcal{V}_R^* = \mathcal{V}_R \cup \mathcal{V}_R^{\text{aux}},$$

where the auxiliary set $\mathcal{V}_R^{\text{aux}}$ is chosen so that every vertex in $\mathcal{V}_R$ lies at least twice the Matérn range parameter away from the outer boundary of $\mathcal{V}_R^*$.

On this extended mesh we construct the SPDE-based precision matrix $\boldsymbol{Q}_{k_i}^*$ as in (10), imposing Neumann boundary conditions on the outer boundary of $\mathcal{V}_R^*$. We then partition $\boldsymbol{Q}_{k_i}^*$ according to the vertices in $\mathcal{V}_R$ and the auxiliary vertices in $\mathcal{V}_R^{\text{aux}}$,

$$\boldsymbol{Q}_{k_i}^* = \begin{pmatrix} \boldsymbol{Q}^*_{\mathcal{V}_R, \mathcal{V}_R} & \boldsymbol{Q}^*_{\mathcal{V}_R, \mathcal{V}_R^{\text{aux}}} \\ \boldsymbol{Q}^*_{\mathcal{V}_R^{\text{aux}}, \mathcal{V}_R} & \boldsymbol{Q}^*_{\mathcal{V}_R^{\text{aux}}, \mathcal{V}_R^{\text{aux}}} \end{pmatrix}.$$

The precision matrix corresponding to the SPDE solution restricted to the interior vertices $\mathcal{V}_R$ is obtained by marginalising out the auxiliary vertices, resulting in the Schur complement

$$\boldsymbol{Q}_{k_i} = \boldsymbol{Q}^*_{\mathcal{V}_R, \mathcal{V}_R} - \boldsymbol{Q}^*_{\mathcal{V}_R, \mathcal{V}_R^{\text{aux}}} \left(\boldsymbol{Q}^*_{\mathcal{V}_R^{\text{aux}}, \mathcal{V}_R^{\text{aux}}} \right)^{-1} \boldsymbol{Q}^*_{\mathcal{V}_R^{\text{aux}}, \mathcal{V}_R}.$$

This matrix is the exact marginal precision of the SPDE solution restricted to the domain of interest. Although $\boldsymbol{Q}_{k_i}$ is dense, its dimension is only $R \ll m$ thanks to the low-rank construction, so the resulting computational cost remains negligible within the overall LR-SSM framework.

4. Simulations

In this section, we evaluate the finite-sample performance of the LR-SSM. To this end, we simulate the observed data using model (1)–(2) with the parameter vector denoted by Π^0, thereby obtaining a continuous latent GP. The inference, instead, is carried out using the LR-SSM (6)–(7). Indeed, Theorem 2 shows that the approximation error when using model (6)–(7) for inference on a data-generating process of model (1)–(2) remains uniformly bounded in time. We consider a realistic setup in which the observed process is of dimension $p = 3$ and the latent space is of dimension $q = 2$, allowing for a fully heterotopic case.

4.1. Experimental design

To study the identifiability of the model defined by (6)–(7), parametrised by Π^0, synthetic data are generated using the SSM defined by Eqs. (1)–(2). Let

$$\mathcal{L}_\delta = \left\{ (i\delta, j\delta) \in \mathbb{R}^2 : i, j \in \{0, 1, \dots, 24\} \right\},$$

denote a 25×25 regular lattice on the domain $\Omega = [0, 1]^2$, with spacing $\delta = 1/25$ between adjacent points. For each variable $i \in \{1, \dots, p\}$, the training set $\mathcal{S}_i \subset \mathcal{L}_\delta$ is constructed by selecting a random subset of m locations. The remaining points, $\mathcal{S}_i^* = \mathcal{L}_\delta \setminus \mathcal{S}_i$, are used to evaluate the model's predictive performance.

We consider three different cases, corresponding to $m = 100$, 225, and 400 spatial locations for each variable. These choices ensure that the simulated sampling regimes are comparable to those encountered in the real-data application, where the heterotopic structure leads to an effective number of observations per component that is similar in magnitude to the simulated values of m. The LR is set to 100%, 75%, 50%, 25% and 15%. As an example, Fig. 2 shows the generated finite element mesh for the simulations when $m = 100$. Except for the case $LR = 100\%$, all other cases use Laplacian smoothing of the Delaunay triangulation. Details on Laplacian smoothing and boundary effects are in Sections 3.1 and 3.2. Moreover, we consider the effect of time-series length, with $T = 50$ and 100.

Summing up, for each rank R, the simulation campaign is based on six different simulation setups. For each simulation setup, $M = 100$ Monte Carlo replications are performed.

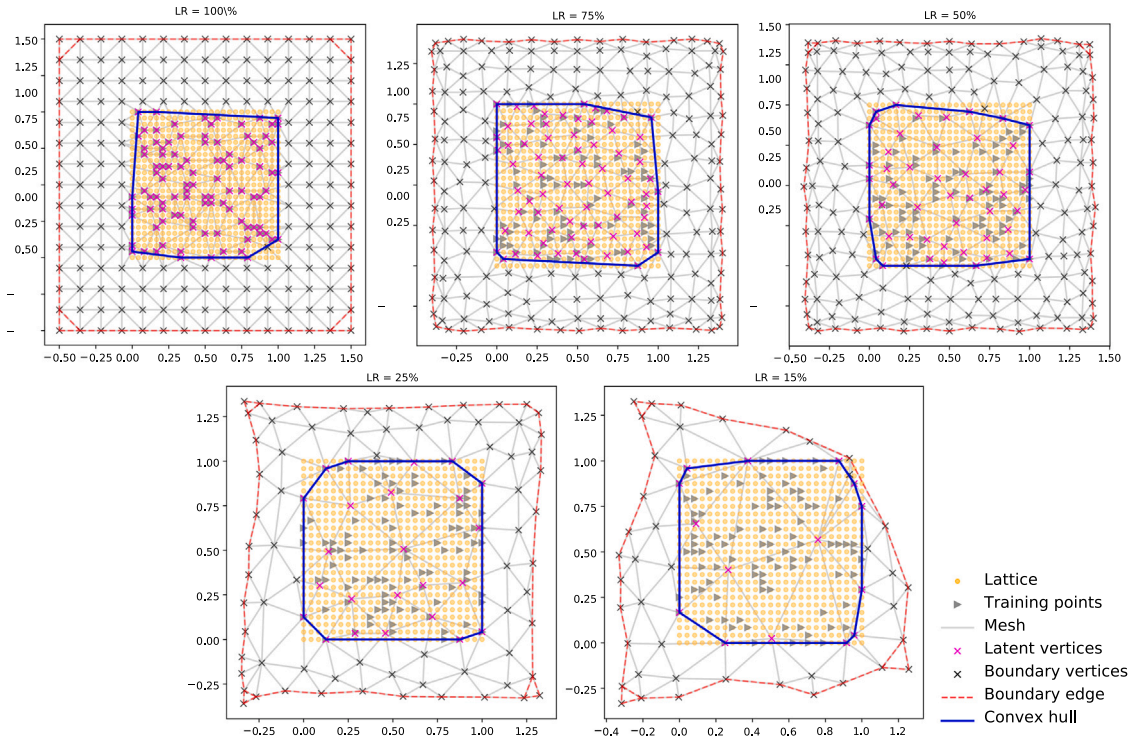


Fig. 2. Simulated $m = 100$ spatial points location on the domain $\Omega = [0,1]^2$, along with the resulting finite element mesh for $LR = 100\%, 75\%, 50\%, 25\%$ and 15% (from left to right, top to bottom). Laplacian smoothing is performed, providing a refined set of vertices (magenta crosses) of the original Delaunay triangulation of the observed points (light grey triangles). The predictive performances are computed on a regular 25×25 lattice $\mathcal{L}_\delta$ (light orange points). The observed points are shown in light grey. The boundary vertices are shown in grey.

The data-generating process is specified by $\Pi^0 = \{\boldsymbol{\beta}^0, \sigma^{2,0}, \boldsymbol{f}^0, \boldsymbol{W}^0, k_1^0, \dots, k_q^0\}$ with the following parameter values: $\boldsymbol{\beta}^0 = (1, 2, -1)'$, and

$$\boldsymbol{X}'(\boldsymbol{s}, t) = \begin{bmatrix} x_1(\boldsymbol{s}, t) & 0 & 0 \\ 0 & x_2(\boldsymbol{s}, t) & 0 \\ 0 & 0 & x_3(\boldsymbol{s}, t) \end{bmatrix},$$

where each $x_i(\boldsymbol{s}, t)$ is independently drawn from a standard normal distribution $\mathcal{N}(0, 1)$, for all spatial locations $\boldsymbol{s} \in \mathcal{L}_\delta$, time steps $t = 1, \dots, T$, and for $i = 1, 2, 3$. The variance parameters are set to $\boldsymbol{\sigma}^{2,0} = (0.5, 1.5, 1)'$. The autoregressive coefficients are defined as $\boldsymbol{f} = (0.85, -0.5)'$, and the $p \times q$ interaction matrix is given by

$$\boldsymbol{W}^0 = \begin{bmatrix} 0.5 & 1 \\ 0.5 & 0.25 \\ 0.2 & 0.8 \end{bmatrix}.$$

The spatial range parameters are set to $\boldsymbol{\kappa}^0 = (\kappa_1^0, \kappa_2^0)' = (7\sqrt{8}, 2\sqrt{8})'$. The latent variables $\boldsymbol{z}(\boldsymbol{s}, t)$ are simulated on the same spatial grid with initial state drawn from $\boldsymbol{z}(\boldsymbol{s}, 0) \sim \mathcal{N}(1, 1)$.

4.2. Estimation procedure

Once a single dataset j, for $j = 1, \dots, M$, is simulated, the EM algorithm is executed with particular starting values. Specifically, as starting values for the estimation procedure, we use a random point uniformly drawn from $[0,1]^2$ for $\boldsymbol{\beta}$, a random point in $(0.2, 0.8) \times (-0.2, -0.8)$ for $\boldsymbol{f}$, each element of $\boldsymbol{W}$ is sampled independently from $\mathcal{U}[0.2, 2]$, each κ_i is drawn from $\mathcal{U}(\sqrt{8}, 7\sqrt{8})$, and each σ_i^2 from $\mathcal{U}[0.1, 2]$.

The maximum iteration number of the EM algorithm is set to 300, while the exit condition is based on the convergence criteria $|\text{logL}_{k+1} - \text{logL}_k| / |\text{logL}_k| < 0.0001$ where logL_k denote the log-likelihood at iteration k.

The implementation of the algorithm is carried out in Python and relies on the NumPy (Harris et al., 2020), SciPy (Virtanen et al., 2020), and JAX (Bradbury et al., 2018) packages for efficient numerical computation. Finite element calculations are performed using the MFEM library (Andrej et al., 2024), which provides a flexible, high-performance framework for assembling variational

Table 1
$LR = 100\%$. Bias and RMSE of the estimated parameters are reported for different numbers of time steps T and spatial locations m. $RMSE_{\text{train}}$ and $RMSE_{\text{test}}$ denote the average prediction RMSEs computed on the training set S_i and the test set S_i^*, respectively, across $M = 100$ Monte Carlo replications. *Avg* and *Std* denote the average and standard deviation, respectively, across replications. Min angle indicates the average of the minimum internal angles across all the Monte Carlo simulations, while Time indicates the estimation time over replications.

Parameter	T = 50						T = 100					
	m = 100		m = 225		m = 400		m = 100		m = 225		m = 400	
	Bias	RMSE	Bias	RMSE	Bias	RMSE	Bias	RMSE	Bias	RMSE	Bias	RMSE
β_1	0.00		0.00		0.00		0.00		0.00		0.00	
β_2	0.00	0.02	0.00	0.02	0.00	0.01	0.00	0.02	0.00	0.01	0.00	0.01
β_3	−0.01		0.00		0.00		0.00		0.00		0.00	
σ_1^2	0.13		0.06		0.02		0.12		0.06		0.02	
σ_2^2	0.28	0.20	0.16	0.11	0.09	0.06	0.28	0.19	0.15	0.10	0.08	0.05
σ_3^2	0.12		0.06		0.02		0.11		0.05		0.02	
k_1	−0.77	2.27	−2.10	3.44	−3.50	3.05	−0.71	2.23	−2.42	2.49	−3.21	2.94
k_2	−0.21		−0.63		−0.81		−0.33		−0.51		−0.79	
W_1	0.05		0.03		0.05		0.09		0.06		0.06	
W_2	−0.03		0.00		0.03		0.00		0.02		0.07	
W_3	−0.01	0.12	−0.05	0.15	−0.10	0.21	0.03	0.12	−0.02	0.15	−0.07	0.21
W_4	−0.01		0.04		0.14		0.00		0.06		0.15	
W_5	0.05		0.06		0.11		0.07		0.08		0.10	
W_6	−0.04		−0.04		−0.06		−0.02		−0.02		−0.02	
f_1	−0.04	0.10	−0.05	0.09	−0.09	0.14	−0.05	0.08	−0.06	0.08	−0.08	0.12
f_2	−0.02		−0.02		−0.01		0.00		0.00		0.00	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std
$\text{RMSE}_{\text{train}}$	1.02	0.02	0.99	0.01	0.97	0.01	1.02	0.01	0.98	0.01	0.96	0.01
RMSE_{test}	1.22	0.01	1.17	0.01	1.13	0.01	1.21	0.01	1.15	0.01	1.12	0.01
Min angle (°)	9.40	1.25	10.51	0.82	10.86	0.26	9.27	1.20	10.53	0.75	10.85	0.28
Time (s)	88.47	25.64	276.32	84.47	818.33	378.96	99.70	37.58	366.46	120.32	1242.62	574.37

forms. Through MFEM's abstractions, standard differential operators, such as the gradient, divergence, and Laplacian, are assembled efficiently, enabling the construction of stiffness and mass matrices in a unified and scalable manner. Mesh quality diagnostics, such as the distribution of minimum internal angles, are essential for evaluating the robustness of our meshing procedure and, consequently, the stability of the finite element discretisation. In the current implementation, mesh quality is monitored by imposing a lower bound on the minimum internal angle of each triangle in the mesh. The initial triangulation is generated using the SciPy Delaunay routine, while the Laplacian smoothing step has been implemented manually to improve mesh regularity.

4.3. Simulation results

We evaluate the models by comparing their parameter estimates using the bias and root mean square error (RMSE) where for each $\theta \in \Pi$ the RMSE is computed as $\sqrt{N^{-1}\sum_{i=1}^{M}\|\hat{\theta}^i - \theta_0\|^2}$ where $\hat{\theta}^i$ corresponds to the estimation of the parameter θ in the ith Monte Carlo simulation, $\|\cdot\|$ is the euclidean norm and θ_0 is the true parameter value. In addition, we assess predictive performance on the validation set S_i^* for $i = 1, \dots, p$, by comparing the observed process $y_i(s,t)$ with its estimate $\hat{y}_i(s,t)$. Prediction errors are defined as $e_i(s,t) = y_i(s,t) - \hat{y}_i(s,t)$, for $s \in S_i^*$. The different LR-SSM versions are then compared in terms of RMSE computed over these validation errors.

Tables 1–3 presents the bias and RMSE of the estimated parameters under each combination of m and T for $LR = 100\%$, 75% and 15% respectively (the simulation results for the cases with $LR = 75\%$ and $LR = 25\%$ are showed in the Supplementary material). First, it has to be remarked that, for all the approximate solutions, the estimation procedure arrived at convergence in all cases, confirming the good stability properties of the EM algorithm for the LR-SSM model. The results indicate that estimation accuracy improves as the number of spatial locations and time steps increases, as evidenced by decreasing RMSE and absolute bias values across all parameters. The ML estimator of $\boldsymbol{\beta}$ has an RMSE close to zero, improving as observed data increase in both T and m. The noise variances $\boldsymbol{\sigma}^2$ show more pronounced improvement with increasing m. Spatial range parameters $\boldsymbol{\kappa}$ are more challenging to estimate accurately; although RMSE decreases with finer grids, a noticeable bias persists. This is a known issue in spatial statistics, as range parameters often exhibit weak identifiability, particularly when data are limited in spatial resolution or extent (Zhang, 2004). The ML estimator of $\boldsymbol{f}$ has small RMSE, and the accuracy improves with increasing time length T. Regarding the accuracy of the estimate of $\boldsymbol{W}$, we observe a small bias that is consistent with the issues in estimating the parameter $\boldsymbol{\kappa}$, which suffers from low identifiability. Indeed, the loading matrix tends to compensate for the effect of the range parameter. Despite the bias affecting these parameters, the training RMSE improves, and the model also shows better performance in terms of test RMSE. (See Table 3.)

Overall, as expected for the ML estimation, these results demonstrate that the LR-SSM yields consistent and increasingly accurate parameter estimates as more spatial and temporal information becomes available, confirming the robustness of the estimation

Table 2
$LR = 50\%$. Bias and RMSE of the estimated parameters are reported for different numbers of time steps T and spatial locations m. $RMSE_{\text{train}}$ and $RMSE_{\text{test}}$ denote the average prediction RMSEs computed on the training set S_i and the test set S_i^*, respectively, across $M = 100$ Monte Carlo replications. *Avg* and *Std* denote the average and standard deviation, respectively, across replications. Min angle indicates the average of the minimum internal angles across all the Monte Carlo simulations, while Time indicates the estimation time over replications.

Parameter	T = 50						T = 100					
	m = 100		m = 225		m = 400		m = 100		m = 225		m = 400	
	Bias	RMSE	Bias	RMSE	Bias	RMSE	Bias	RMSE	Bias	RMSE	Bias	RMSE
β_1	0.00		0.00		0.00		0.00		0.00		0.00	
β_2	0.00	0.03	0.00	0.02	0.00	0.01	0.00	0.02	0.00	0.01	0.00	0.01
β_3	0.00		0.00		0.00		0.00		0.00		0.00	
σ_1^2	0.42		0.27		0.21		0.41		0.26		0.19	
σ_2^2	0.32	0.32	0.22	0.21	0.16	0.16	0.34	0.32	0.23	0.20	0.15	0.14
σ_3^2	0.15		0.09		0.05		0.15		0.08		0.05	
k_1	2.63	4.94	−0.11	3.48	−4.01	4.50	3.02	3.95	0.93	4.70	−3.43	3.52
k_2	0.65		−0.55		−0.65		0.19		−0.60		−0.59	
W_1	0.15		0.08		0.09		0.16		0.07		0.10	
W_2	−0.01		−0.01		0.01		0.03		0.05		0.03	
W_3	0.11	0.15	−0.02	0.18	−0.08	0.22	0.13	0.15	0.03	0.16	−0.04	0.22
W_4	0.00		0.07		0.15		0.01		0.06		0.13	
W_5	0.11		0.12		0.15		0.11		0.09		0.14	
W_6	−0.02		−0.05		−0.08		0.01		0.02		−0.05	
f_1	−0.06	0.11	−0.06	0.11	−0.09	0.12	−0.04	0.08	−0.05	0.09	−0.09	0.13
f_2	−0.01		−0.02		−0.04		0.01		−0.01		−0.02	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std
$\text{RMSE}_{\text{train}}$	1.09	0.01	1.05	0.01	1.02	0.01	1.08	0.01	1.04	0.01	1.02	0.01
$\text{RMSE}_{\text{test}}$	1.22	0.01	1.17	0.01	1.13	0.01	1.22	0.01	1.16	0.00	1.12	0.01
Min angle (°)	11.40	3.50	10.67	1.12	10.68	0.67	11.74	3.48	10.61	0.98	10.74	0.56
Time (s)	36.42	8.61	101.50	19.31	273.86	76.99	33.63	8.23	130.17	35.38	373.08	156.46

Table 3
$LR = 15\%$. Bias and RMSE of the estimated parameters are reported for different numbers of time steps T and spatial locations m. $RMSE_{\text{train}}$ and $RMSE_{\text{test}}$ denote the average prediction RMSEs computed on the training set S_i and the test set S_i^*, respectively, across $M = 100$ Monte Carlo replications. *Avg* and *Std* denote the average and standard deviation, respectively, across replications. Min angle indicates the average of the minimum internal angles across all the Monte Carlo simulations, while Time indicates the estimation time over replications.

Parameter	T = 50						T = 100					
	m = 100		m = 225		m = 400		m = 100		m = 225		m = 400	
	Bias	RMSE	Bias	RMSE	Bias	RMSE	Bias	RMSE	Bias	RMSE	Bias	RMSE
β_1	0.00		0.00		0.00		0.00		0.00		0.00	
β_2	0.01	0.03	0.00	0.02	0.00	0.01	0.00	0.02	0.00	0.01	0.00	0.01
β_2	0.00		0.00		0.00		0.00		0.00		0.00	
σ_1^2	0.76		0.54		0.41		0.75		0.52		0.39	
σ_2^2	0.52	0.56	0.39	0.40	0.31	0.31	0.54	0.56	0.40	0.39	0.30	0.29
σ_3^2	0.27		0.17		0.13		0.26		0.16		0.11	
k_1	−0.27	3.12	4.16	5.50	4.80	7.49	0.23	2.35	4.05	5.63	5.55	6.92
k_2	0.86		0.68		0.42		0.72		0.34		0.42	
W_1	0.77		0.57		0.45		0.86		0.59		0.43	
W_2	0.08		0.13		0.15		0.10		0.21		0.19	
W_3	0.34	0.47	0.29	0.38	0.19	0.37	0.39	0.51	0.31	0.41	0.21	0.35
W_4	0.05		0.12		0.19		0.05		0.16		0.19	
W_5	0.57		0.42		0.35		0.64		0.42		0.32	
W_6	0.06		0.07		0.05		0.07		0.12		0.09	
f_1	−0.37	0.31	−0.22	0.19	−0.18	0.18	−0.41	0.33	−0.21	0.20	−0.17	0.17
f_2	0.08		0.06		0.01		0.08		0.07		0.03	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std
$\text{RMSE}_{\text{train}}$	1.20	0.01	1.14	0.01	1.10	0.01	1.20	0.01	1.14	0.01	1.10	0.01
$\text{RMSE}_{\text{test}}$	1.27	0.01	1.21	0.01	1.16	0.01	1.27	0.01	1.20	0.01	1.16	0.01
Min angle (°)	19.34	3.88	15.70	4.47	13.98	4.56	19.57	3.61	15.79	4.39	13.75	4.22
Time (s)	14.07	5.55	23.27	6.98	59.05	22.03	11.46	4.02	28.12	9.03	81.01	31.13

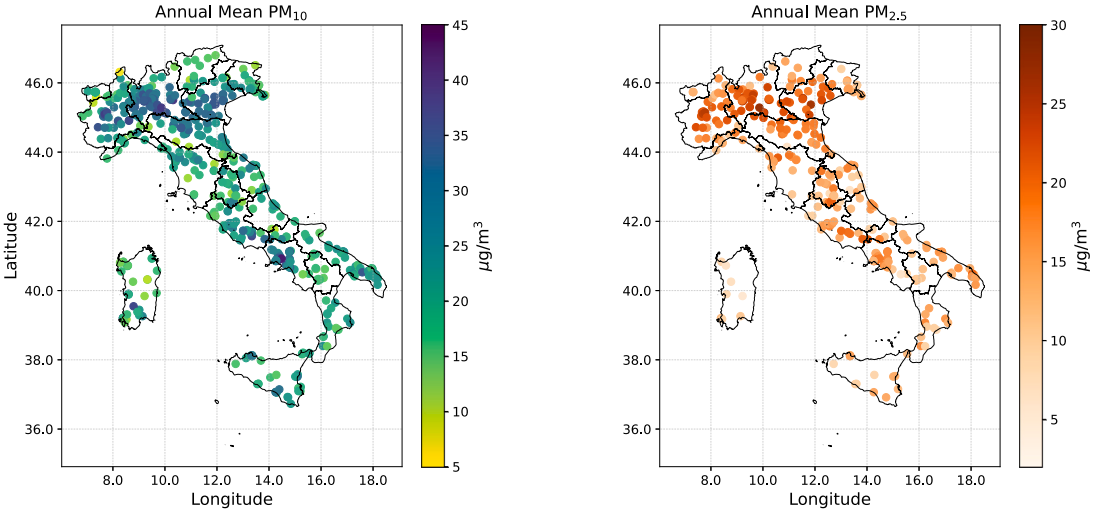


Fig. 3. Monitoring network and overall averages of PM_{10} (left) and $PM_{2.5}$(right) concentration over Italy. A total of $S = 661$ stations are available: $S_1 = 574$ stations measuring PM_{10}, and $S_2 = 313$ measuring $PM_{2.5}$, with 304 stations providing co-located measurements.

procedure under increasing low-rank approximation. This means that, in general, the low-rank approximation can lead to a speed-up in computation without losing precision. Furthermore, the estimate indicates that the model performs well in the heterotopic case, meaning it can estimate the cross-correlation among the observed processes. By evaluating the test RMSE under increasing levels of low-rank approximation R, while keeping m fixed, we observe a slight increase in RMSE that is not statistically significant compared to the case $LR = 100\%$. This demonstrates that the approximation procedure substantially reduces computation time while maintaining consistent parameter estimates and accurate spatial mapping performance. The computation time refers to a machine equipped with an Intel Xeon Platinum 8460Y+ CPU and 1007 GB RAM using Python 3.10.15.

5. Analysis of a multivariate spatio-temporal air quality dataset

We now illustrate how the model is used to analyse a large spatio-temporal dataset with multiple pollutants, heterogeneous spatial coverage, and substantial missingness.

We focus on two pollutants: coarse particulate matter with aerodynamic diameter less than 10 μm (PM_{10}) and fine particulate matter with diameter less than 2.5 μm ($PM_{2.5}$) in Italy. These pollutants are typically observed at partially different monitoring stations (heterotopic) and are known to be highly correlated. Fassò et al. (2016) shows that the estimated correlation between co-located PM_{10} and $PM_{2.5}$ at the European level is close to 0.93 after controlling for meteorology and orography. Hence, leveraging this correlation, the aim is to obtain maps of daily concentrations at the national level. This requires adopting a statistical model capable of handling the complexity of the bivariate pollutant phenomenon, while controlling for meteorology and orography, as well as accounting for an unbalanced monitoring network, heterotopic observations, and extensive missing data.

5.1. Data description

We use the AQCLIM-GRINS dataset provided by Fusta Moro and Fassò (2025), which contains daily air quality and meteorological measurements across Italy, covering the five-and-a-half-year period from July 1, 2018, to December 30, 2023. The AQCLIM-GRINS dataset extends the AGRIMONIA dataset (Fassò et al., 2023) to the entire Italian territory and has been harmonised by the GRINS project (https://www.grins.it/, accessed 26/07/2025).

The particulate matter data consists of daily average concentrations (in $\mu g/m^3$) collected at 661 ground-level monitoring stations irregularly distributed across Italy. Also, we include the following covariates: altitude (in metres above sea level), temperature at 2 metres (in °C), relative humidity (RH; in %), and wind Speed (in m/s), along with an intercept term. Table 4 summarises the variables used in this study and the associated main statistics.

It can be noticed that the resulting measurement network in Fig. 3 is partially heterotopic, meaning the variables are observed at intersecting but non-identical sets of locations. The considered period spans $T = 2009$ days, and the dataset includes approximately 1.15 million observations of PM_{10} from 574 monitoring stations, with 28.7% missing values, and about 0.6 million observations of $PM_{2.5}$ from 313 stations, with 34.1% missing values. Concentrations above the 99.9 percentile are treated as outliers and replaced with a missing value.

Table 4
Descriptive statistics of air quality and meteorological variables from the AQCLIM-GRINS dataset. Abbreviations: Alt.: altitude; Temp.: temperature; RH: relative humidity; WS: wind speed; SD: standard deviation.

Statistics	PM_{10} ($\mu g/m^3$)	$PM_{2.5}$ ($\mu g/m^3$)	Alt. (m)	Temp. (°C)	RH (%)	WS (m/s)
Percentiles						
Min.	0.00	0.00	−2.00	−24.68	15.27	0.11
5th	7.00	4.00	4.00	1.63	52.95	0.71
25th	13.76	8.00	26.00	8.31	66.59	1.18
50th	20.00	12.00	114.00	13.84	75.70	1.64
75th	29.55	19.00	271.00	20.29	83.63	2.45
95th	56.80	41.40	736.00	26.32	92.01	4.64
Max.	137.40	102.60	1770.00	35.75	99.98	16.54
Descriptive statistics						
Mean	24.14	15.66	209.31	14.03	74.51	2.02
SD	16.20	12.49	274.34	7.84	11.99	1.29
Skewness	1.98	2.28	2.43	−0.14	−0.49	2.02
Kurtosis	5.69	6.87	7.61	−0.47	−0.12	5.82
Number of stations	574	313	661	661	661	661
Missing proportion	0.29	0.34	0.00	0.00	0.00	0.00

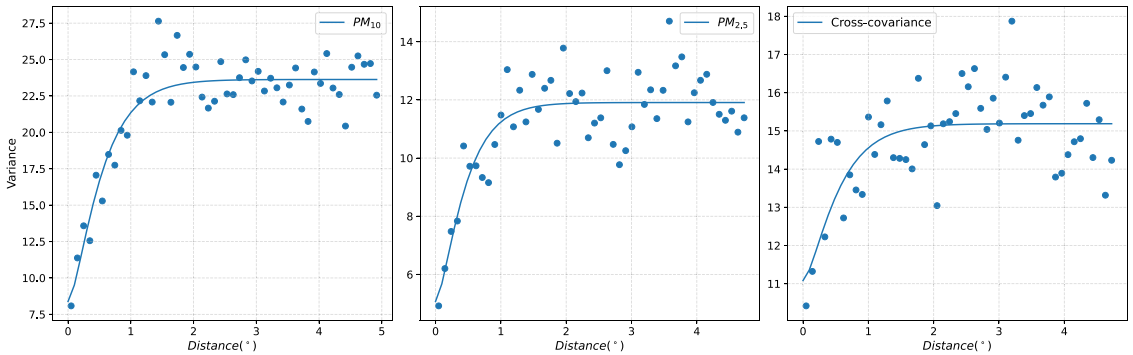


Fig. 4. Spatial variograms and covariogram computed from the average concentration at each spatial location in $(\mu g/m^3)^2$. The plots show the variogram of PM_{10} (left), the variogram of $PM_{2.5}$ (centre), and the cross-covariogram between PM_{10} and $PM_{2.5}$ (right). The fitted Matérn models are estimated with weighted least squares (Cressie, 1985).

5.2. Preliminary analysis

Fig. 3 highlights spatial patterns in pollutant average concentrations: nearby stations tend to record similar values, suggesting the presence of spatial autocorrelation and cross-correlation between the two particulate matter variables. This is confirmed by Fig. 4 reporting the weighted least squares Matérn variograms and covariogram of station averages for PM_{10} and $PM_{2.5}$ (Wackernagel, 2003; Cressie, 1985). The variograms reveal a nugget effect, with estimated measurement error std $\hat{\sigma}_i$ of 2.88 $\mu g/m^3$ for PM_{10} and 2.24 $\mu g/m^3$ for $PM_{2.5}$. Accordingly, the model of the next section includes a measurement error component as specified in Eq. (7). The estimated marginal variances of the spatial process are approximately 15.4 and 6.8 $(\mu g/m^3)^2$. As expected, the autocovariances of the two detrended processes decrease with distance. The estimated spatial ranges are about 0.37 ° (41 km) and 0.30 ° (33 km), respectively for PM_{10} and $PM_{2.5}$. Similarly, the correlation between PM_{10} and $PM_{2.5}$ is particularly high, see Fig. 4 (right panel), and has a spatial range of 0.37 ° (41 km), which is close to the PM_{10} behaviour since networks are almost overlapping.

The temporal variation of the observed PM_{10} and $PM_{2.5}$ concentrations is shown as daily averages among the sites in Fig. 5 (left panel). Not surprisingly, there is an apparent seasonality with higher concentrations in winter. Moreover, the empirical autocorrelation function (ACF) is shown in Fig. 5. The first lag is close to 0.7 for both pollutants, indicating temporal dependence. Time series and ACF grouped by regions are summarised in the Supplementary material.

5.3. Application of the LR-SSM

To apply the LR-SSM (6)–(7) defined in Section 2.1, we discretise the considered domain using a Delaunay triangulation to form the spatial meshes $\mathcal{V}_R^j$ for $j = 1, 2$. This triangulation maximises the smallest angle among all triangles, promoting mesh regularity. To further ensure good mesh quality, we apply the Laplacian smoothing with a minimum internal angle threshold of 0.15 °. Additional details on the triangulation and smoothing procedures are given in Section 3.1. The resulting discretised domain is shown in Fig. 6.

We validate the model using a data split validation strategy tailored to the spatio-temporal setting (see Otto et al., 2024, for a review). Specifically, we adopt a leave-many-stations-out validation scheme. Basically, the entire time series of a subset of stations

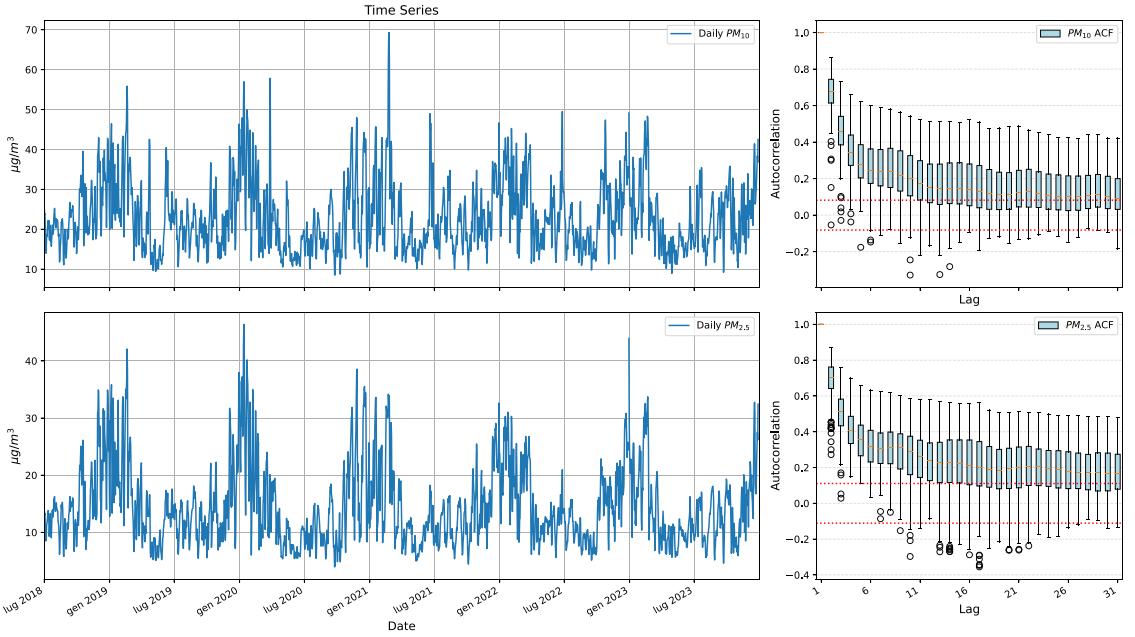


Fig. 5. Temporal structure of particulate matter concentrations. The left column shows daily averages of PM_{10} (top) and $PM_{2.5}$ (bottom); the right column displays corresponding autocorrelation functions (ACFs) over the first 30 lags, summarised across stations using boxplots.

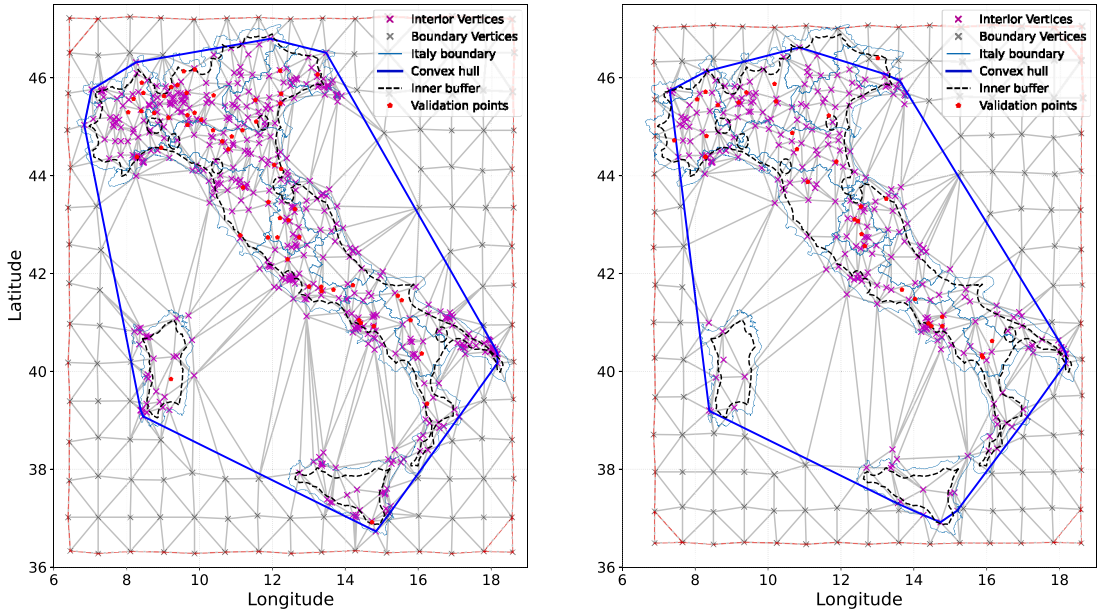


Fig. 6. Spatial meshes generated via Laplacian smoothing. The left panel corresponds to the mesh constructed for the PM_{10} dataset, and the right panel to the $PM_{2.5}$ dataset. Interior vertices (magenta), boundary vertices (green), convex hulls (blue), inner buffer (black), and validation locations (red) are shown.

(the validation set) is withheld during training and later used to assess the model's predictive performance. We construct the validation set by randomly selecting 10% of the total stations for each pollutant, while the remaining 90% are used for training. To avoid edge effects, we exclude stations close to the domain edge by selecting only stations located at least 0.15° away from the boundary (see the inner buffer in Fig. 6). This choice is motivated by the need to avoid potential boundary-related artefacts driven by domain truncation rather than by model misspecification, which are not of primary interest for the present modelling objectives.

Table 5
Prediction accuracy and estimation time of LR-SSMs with varying low-rank (LR) percentages on the AQCLIM-GRINS dataset. Results are reported for both the training and validation sets in terms of RMSE and R^2, along with the total estimation time and the number of EM iterations. R denotes the number of vertices in the spatial mesh used for each model component. "Time (s)" refers to the total model estimation time in seconds, and "EM iter." indicates the number of iterations required for model estimation.

LR	Variable	R	Train		Validation		Time (s)	EM iter.
			RMSE	R^2	RMSE	R^2		
100%	PM_{10}	516	3.64	0.94	6.92	0.80	12,360	43
	$PM_{2.5}$	281	2.09	0.96	5.84	0.77		
75%	PM_{10}	387	4.17	0.92	6.97	0.80	6,120	35
	$PM_{2.5}$	211	2.77	0.94	5.94	0.76		
50%	PM_{10}	258	4.98	0.89	6.70	0.82	3,420	33
	$PM_{2.5}$	141	3.58	0.90	6.10	0.75		
25%	PM_{10}	129	6.01	0.84	6.88	0.81	780	25
	$PM_{2.5}$	71	4.31	0.85	6.69	0.70		

Table 6
ML estimates of regression coefficients and random effect parameters for the bivariate model of PM_{10} and $PM_{2.5}$, for different low-rank approximation levels (LR = 100%, 75%, 50% and 25%). Covariates include altitude, temperature at 2 metres, relative humidity, and wind speed. Values are followed by their standard deviations in parentheses.

Parameter	$LR = 100\%$		$LR = 75\%$		$LR = 50\%$		$LR = 25\%$	
	PM_{10}	$PM_{2.5}$	PM_{10}	$PM_{2.5}$	PM_{10}	$PM_{2.5}$	PM_{10}	$PM_{2.5}$
Intercept	15.762	12.154	16.809	12.694	16.814	13.145	15.220	12.673
	(0.029)	(0.026)	(0.034)	(0.032)	(0.038)	(0.036)	(0.044)	(0.044)
Altitude	−0.012	−0.007	−0.014	−0.008	−0.012	−0.007	−0.012	−0.009
	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)
Temperature	−0.055	−0.236	−0.060	−0.239	−0.056	−0.257	−0.053	−0.269
	(<0.001)	(<0.001)	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)
RH	0.088	0.062	0.082	0.057	0.074	0.054	0.096	0.069
	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)
Wind Speed	−0.760	−0.874	−0.854	−0.863	−0.788	−0.830	−0.837	−0.926
	(0.003)	(0.003)	(0.003)	(0.003)	(0.004)	(0.004)	(0.004)	(0.004)
W_1	6.998	2.439	6.515	2.575	6.200	2.752	6.008	3.375
	(0.001)	(0.001)	(0.001)	(0.001)	(0.002)	(0.001)	(0.002)	(0.001)
W_2	3.467	3.688	3.351	3.356	3.425	3.198	3.584	3.184
	(0.002)	(0.001)	(0.002)	(0.001)	(0.002)	(0.001)	(0.002)	(0.002)
σ_1^2	18.535	7.825	24.642	11.633	30.808	15.107	41.300	21.828
	(0.008)	(0.004)	(0.012)	(0.008)	(0.016)	(0.011)	(0.025)	(0.020)
f	0.823	0.869	0.827	0.889	0.844	0.903	0.895	0.946
	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)	(<0.001)
k	3.349	3.232	2.676	2.197	2.118	1.844	1.578	1.904
	(0.002)	(0.003)	(0.002)	(0.002)	(0.002)	(0.003)	(0.002)	(0.004)

The resulting training set includes 516 and 281 spatial points, while the validation set includes 58 and 32 spatial points for PM_{10} and $PM_{2.5}$, respectively.

To validate the low-rank approach and given a set of candidate R values, the mean square error optimality is of interest. Hence, considering $\hat{y}_i(\boldsymbol{s},t)$ as in (8), errors $e_i(\boldsymbol{s},t) = y_i(\boldsymbol{s},t) - \hat{y}_i(\boldsymbol{s},t)$ with $\boldsymbol{s} \in \Omega$ are computed and the LR-SSM candidates are compared in terms of RMSE and R^2 statistic, where, for each pollutant i, $R_i^2 = 1 - \sum_{t=1}^{T} \sum_{\boldsymbol{s}\in S_{i,t}} (y_i(\boldsymbol{s},t) - \hat{y}_i(\boldsymbol{s},t))^2 / \sum_{t=1}^{T} \sum_{\boldsymbol{s}\in S_{i,t}} (y_i(\boldsymbol{s},t) - \bar{y}_i)^2$, where $\bar{y}_i = T^{-1} \sum_{t=1}^{T} |S_{i,t}|^{-1} \sum_{\boldsymbol{s}\in S_{i,t}} y_i(\boldsymbol{s},t)$.

We consider low-rank approximations where LR is set to 100%, 75%, 50%, and 25%. Table 5 summarises the validation results across models. As expected, for both pollutants, the training RMSE increases monotonically with the degree of approximation, whereas the test RMSE increases only slightly, at most by 6% in the worst case. These results confirm that the low-rank model preserves predictive accuracy while significantly reducing estimation time. Remarkably, even at the roughest approximation level (25% of the rank), the validation RMSE increases by only 6% for PM_{10} and 15% for $PM_{2.5}$ compared to the full-rank model, while estimation time is reduced from 3 h 37 m to just 16 m. This demonstrates that substantial efficiency gains can be achieved without compromising accuracy in a practically meaningful way. Notably, across all low-rank configurations, the test RMSEs remain comparable to those reported in Rodeschini et al. (2024) and Fassò et al. (2016).

Table 6 reports the ML estimates and the associated uncertainty of the parameter vector $\hat{\Pi}_{MLE}$ for different low-rank approximation levels. For both PM_{10} and $PM_{2.5}$, altitude exhibits a small negative association, indicating that higher elevation is associated with lower pollution levels. Temperature shows a negative effect, especially for $PM_{2.5}$. Relative Humidity is positively correlated with both pollutants, more pronounced for PM_{10}. Wind Speed has a substantial negative effect on both processes, with higher winds contributing to pollutant dispersion. The results are coherent with the analysis of Rodeschini et al. (2024).

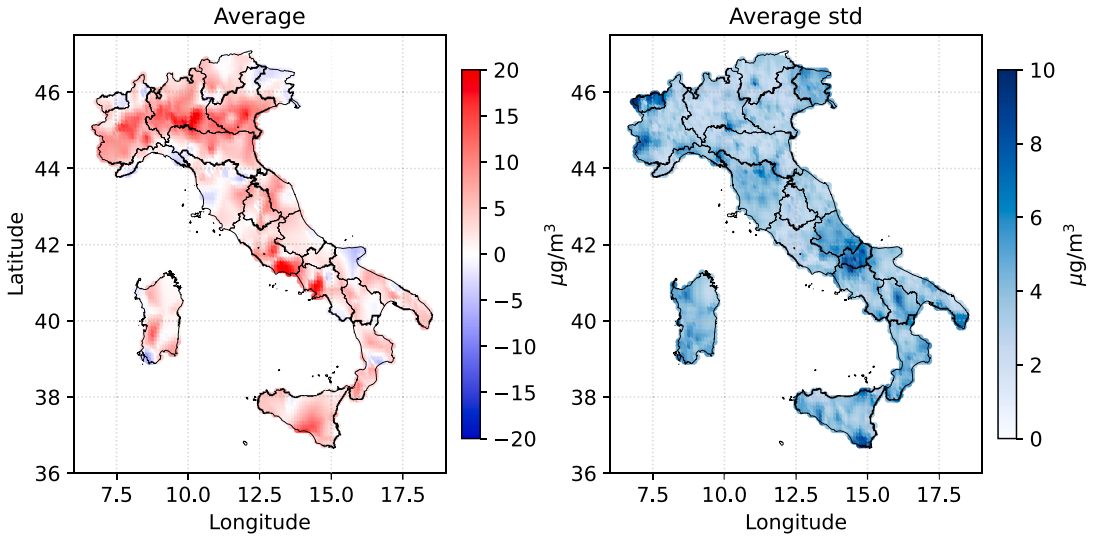


Fig. 7. Estimated average latent effect component on PM_{10} (left) and associated with the average standard deviation (right).

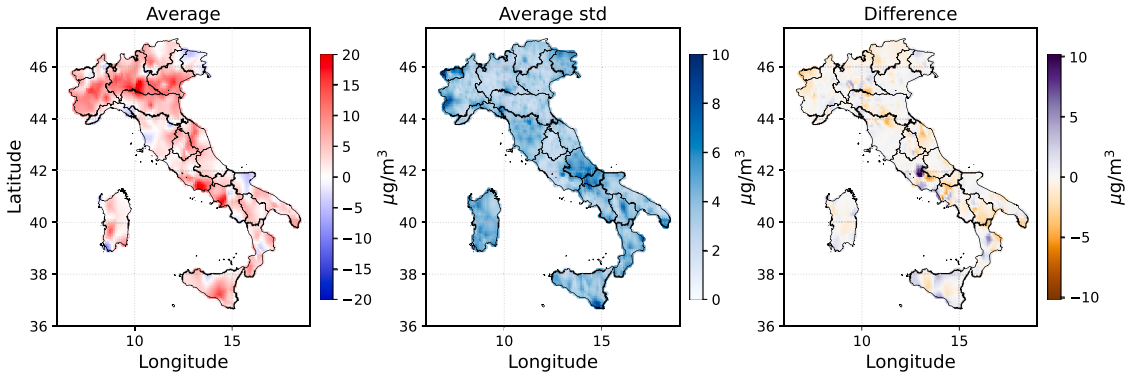


Fig. 8. $LR = 75\%$. Estimated average latent effect component on PM_{10} (left), associated with the average standard deviation (centre) and average difference between $LR = 100\%$ and $LR = 75\%$ estimations (right).

Finally, Figs. 7–10 show the estimated average random effect component of the PM_{10} across the study domain. Each map depicts the mean over all available years, corresponding to the average over time of $\hat{\boldsymbol{W}}\boldsymbol{H}(\boldsymbol{s})\hat{\boldsymbol{z}}_t$ and its standard deviation, as defined in the predictive Eqs. (8) and (9). As expected, prediction performance decreases as the approximation level increases. Nevertheless, the average standard deviation remains low across all cases. The magnitude of the random effects suggests some level of model misspecification, specifically, that in certain areas, the available covariates are insufficient to fully capture the variability of the process. In Fig. 7, the Po Valley in Northern Italy, one of the most polluted regions in Europe, is clearly distinguishable due to its large positive random effects. This result aligns with findings from other studies focusing on Northern Italy, such as Rodeschini et al. (2024). As the approximation level increases (LR decreases), the latent process becomes smoother, which is also reflected in the reduced values of the estimated rescaling parameters k_i. In the approximation case $LR = 25\%$ in Fig. 10, some local variability present in the data is lost. Indeed, the difference from the full-rank case (i.e., $LR = 100\%$) is especially noticeable in regions with a high density of stations but where the mesh is relatively sparse. Regarding the estimated uncertainty, it remains consistent with the test RMSE, which is approximately 6 $\mu g/m^3$. Notably, no boundary effects are observed, and the uncertainty remains stable even near the edges of the domain.

6. Comparison with INLA

In this section, we benchmark the predictive accuracy and computational efficiency of our modelling and model estimation strategy against two strategies based on the SPDE-INLA, considering the AQCLIM-GRINS dataset discussed in Section 5. These strategies are: the LCM discrete-time state–space formulation, which extends (Cameletti et al., 2013) to the bivariate case, and the continuous spatio-temporal field formulation rooted in the diffusion-based spatio-temporal SPDE proposed by Lindgren et al. (2024).

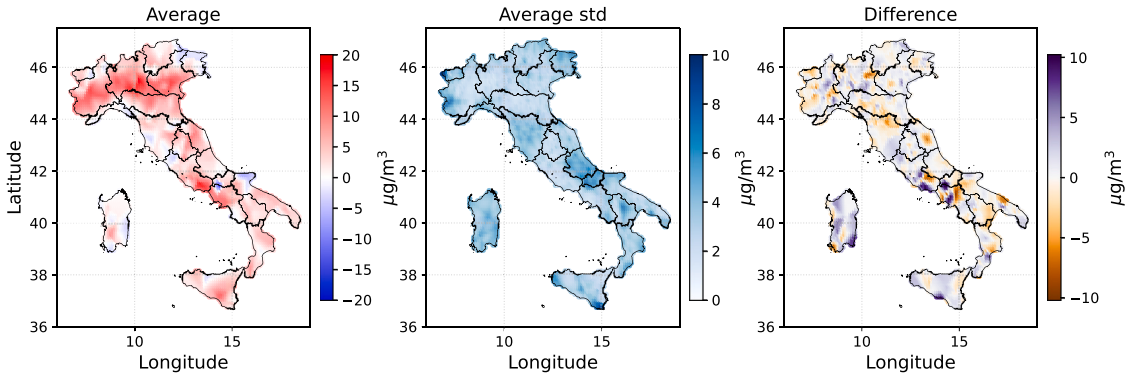


Fig. 9. $LR = 50\%$. Estimated average latent effect component on PM_{10} (left), associated with the average standard deviation (centre) and average difference between $LR = 100\%$ and $LR = 50\%$ estimations (right).

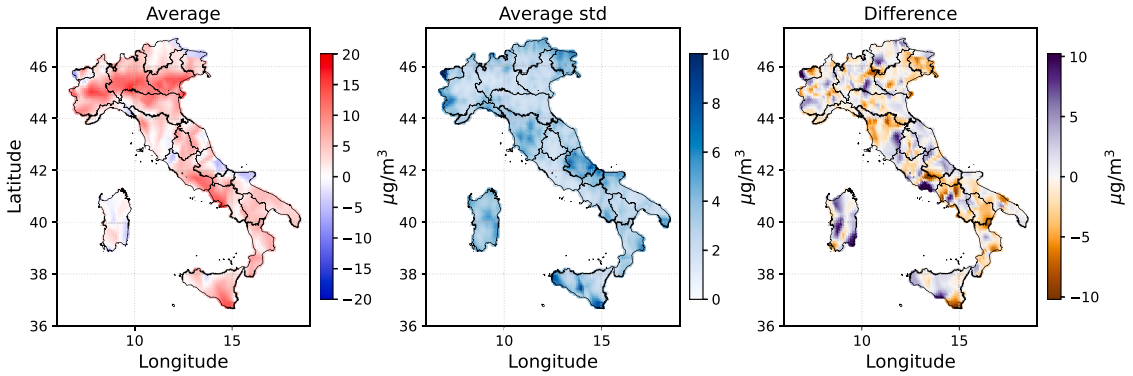


Fig. 10. $LR = 25\%$. Estimated average latent effect component on PM_{10} (left), associated with the average standard deviation (centre) and average difference between $LR = 100\%$ and $LR = 25\%$ estimations (right).

The first strategy serves as a direct counterpart to our proposal. It is formulated as a first-order Markovian state–space model where the latent state satisfies (2) and the spatial field is estimated using the SPDE approximation. Since this specification is structurally identical to our proposed model, it allows us to isolate the impact of the inference method. We denote this strategy as SSM-INLA.

The second strategy defines the latent state as a non-separable continuous spatio-temporal Gaussian random field. It is derived as the solution to a stochastic partial differential equation that combines a Matérn-type spatial operator with a first-order temporal diffusion process. Specifically, we adopt *Model 102* from the taxonomy of Lindgren et al. (2024). We denote this strategy as DIFF-INLA.

We compare these two strategies against the LR-SSM EM framework. Our main interest lies in comparing predictive performance and estimation time. To ensure a fair comparison, we consider similar mesh sizes and restrict the discussion to $LR = 100\%$. Thus, for brevity, we denote our method as SSM-EM.

For all the strategies, we specify two distinct spatial meshes, one for each latent component, constructed from the locations of monitoring stations measuring $PM_{2.5}$ and PM_{10}, respectively. For the INLA-based strategies, meshes are generated using the fmesher library (Lindgren, 2023). For the SSM-EM, we adopt the same mesh construction strategy as in Section 5. Across all methods, the meshes are chosen to have comparable number of vertex, ensuring that differences in predictive performance and computational cost are attributable to the modelling and inference strategies rather than to discrepancies in spatial discretisation. For DIFF-INLA, the spatio-temporal field is represented using a separable construction in which the temporal evolution is discretised via piecewise linear basis functions and coupled with the spatial representation induced by the mesh.

All computations are performed on the same hardware described in Section 4.3. The SSM-INLA strategy is estimated using the `INLA` R package (version 25.12.16), while DIFF-INLA is estimated through the `INLAspacetime` R package (version 0.1.12). To ensure a fair comparison in the estimation, we explicitly control the number of threads available to the INLA internal solver, thereby enforcing equal resource allocation across all strategies.

Regarding prior specification for SSM-INLA, we consider the marginal variance of the latent innovation term fixed to 1 to ensure correspondence with our proposal. The range parameter is assigned a penalised complexity (PC) weakly informative prior such that $\Pr(\rho < 0.5) = 0.5$. The autoregressive coefficients are given the default INLA prior, while the regression coefficients $\boldsymbol{\beta}$, and the

Table 7
Comparative performance analysis on AQCLIM-GRINS dataset subsets. The table reports average training and test RMSE ($\mu g/m^3$) and average model estimation time for: the proposed low-rank state–space model estimated via EM (SSM-EM), the equivalent state–space formulation estimated via INLA (SSM-INLA), and the diffusion-based spatio-temporal field formulation (DIFF-INLA). Estimation time is expressed in seconds and as a speed-up factor relative to SSM-INLA. Averages are computed from 10 independent runs based on randomly selected disjoint temporal windows of length T. Standard deviations are provided in parentheses. Bold values highlight the lowest RMSE for each combination of T, pollutant, and metric.

T	Model	Est.	RMSE train				RMSE test				Estimation time	
			$PM_{2.5}$		PM_{10}		$PM_{2.5}$		PM_{10}		Time (s)	Speed-up
15	SSM	EM	**2.25**	(0.72)	**2.75**	(0.58)	4.90	(1.09)	6.86	(1.44)	185 (29)	1.8×
		INLA	3.68	(1.04)	5.36	(1.13)	3.80	(1.27)	5.78	(1.27)	328 (86)	1.0×
	DIFF	INLA	3.46	(0.84)	4.92	(0.72)	**3.56**	(1.08)	**5.61**	(1.18)	512 (133)	0.6×
30	SSM	EM	**2.44**	(0.71)	**3.11**	(0.53)	5.13	(1.23)	7.09	(1.42)	215 (57)	5.0×
		INLA	3.84	(1.16)	5.43	(1.16)	4.17	(1.46)	6.24	(1.41)	1077 (366)	1.0×
	DIFF	INLA	3.67	(1.01)	5.17	(0.96)	**3.95**	(1.26)	**6.14**	(1.25)	1845 (393)	0.6×
60	SSM	EM	**2.08**	(0.50)	**2.01**	(0.44)	5.31	(0.98)	**4.07**	(0.92)	141 (45)	21.8×
		INLA	3.68	(1.08)	5.17	(1.16)	3.92	(1.36)	6.00	(1.42)	3078 (1048)	1.0×
	DIFF	INLA	3.59	(0.96)	4.94	(0.91)	**3.83**	(1.21)	6.00	(1.32)	5692 (1725)	0.5×

loading matrix $\boldsymbol{W}$ are assigned independent weakly informative Gaussian priors $N(0, 10^3)$. For DIFF-INLA, we adopt PC priors for the spatial range r_s, the temporal range r_t, and the latent field standard deviation σ. We specify $\Pr(r_s < 0.5) = 0.5$ and $\Pr(r_t < 5) = 0.5$, effectively setting the prior medians for the spatial and temporal correlation lengths. Conversely, for the variance, we impose a stricter constraint of $\Pr(\sigma > 1) = 0.01$. This penalises large variances in the latent fields, ensuring that the magnitude of the observed variation is primarily determined by the mixing weights of the LCM, thereby avoiding scale non-identifiability.

A critical limitation of the INLA strategies is that their estimation cost scales poorly with the number of time steps T. While our method efficiently processes the full AQCLIM-GRINS dataset, comprising $T = 2009$ daily observations (≈ 5.5 years), applying the SPDE-INLA strategies appears computationally infeasible. For this reason, we consider time horizons $T \in \{15, 30, 60\}$. For each choice of T, we conduct 10 independent runs, each based on a randomly selected temporal window. The selected windows are mutually disjoint for a given T.

The comparative analysis in Table 7 highlights a contrast in computational efficiency between the EM and the INLA strategies. The most interesting advantage of SSM-EM is its scalability: while the computational burden of INLA implementations grows rapidly with the temporal window T (exceeding three hours for $T = 60$), our approach maintains a stable execution time (under four minutes). Interestingly, as the temporal dimension remains small, the linear increase in time is not visible, since the computation time here mostly depends on the number of iterations of the EM algorithm, and a larger dataset ($T = 60$) may decrease the required number of iterations. This efficient scaling profile makes SSM-EM suitable for high-dimensional spatio-temporal analyses where INLA inference methods become computationally prohibitive without relying on high-performance computing (e.g., Gaedke-Merzhäuser et al., 2025).

In terms of predictive accuracy, SSM-INLA and DIFF-INLA tend to produce the lowest out-of-sample RMSE on average. DIFF-INLA in particular frequently yields the most accurate predictions, albeit at the highest computational cost. The SSM-EM model exhibits high flexibility, achieving an exceptional fit to the observed data as evidenced by the lowest training RMSE. However, this flexibility leads to a divergence between training and test errors at shorter time windows, indicating potential overfitting. This observation is consistent with the fact that $LR = 100\%$ may not always be optimal, as it can overfit the training data relative to a lower LR when the dataset is small. Nevertheless, we observe that predictive accuracy improves and stabilises as the size of the dataset increases. For the longest window, $T = 60$, SSM-EM maintains its computational speed advantage and achieves the best out-of-sample performance for PM_{10}. This suggests that the model effectively captures the underlying latent structure as information increases.

7. Discussion and conclusions

This paper develops a scalable framework for high-dimensional multivariate spatio-temporal data, building on state–space models that leverage the SPDE-GMRF approach for computationally efficient inference. The core contribution lies in the combination of these components, which confers several advantages over existing alternatives. First, the FEM basis provides a model-based, interpretable low-rank representation with controlled approximation error, thereby avoiding the ad hoc basis selection commonly required in other low-rank methods. Second, the resulting sparse structure is embedded within the Kalman filtering framework, preserving linear-time filtering and enabling likelihood-based parameter estimation via an EM algorithm with mostly closed-form updates.

Our proposal naturally supports multivariate heterotopic datasets, while the EM-based estimation procedure efficiently accommodates extensive missing data. We demonstrated weak convergence of the approximated latent field to the target Matérn field as the domain discretisation is refined. For prediction, we established time-independent bound on the prediction RMSE, rigorously quantifying the quality of the SPDE approximation with respect to the time horizon.

We showed that the computational cost can be controlled by the size of the mesh used to discretise the spatial domain, allowing users to balance the trade-off between accuracy and runtime according to available computational resources. Numerical experiments indicate that increasing the approximation level yields prediction errors comparable to those of the SSM without mesh reduction,

while substantially reducing computational demands. Notably, a higher rank does not necessarily imply better performance, as SSMs are flexible models that can easily overfit the training data. Consequently, the selection of the rank remains an open issue, and cross-validation may be required to identify an appropriate value. In a real-case application to bivariate air-quality modelling, the LR-SSM achieves an approximate 16× speed-up in computation time with only a slight increase in prediction error.

To illustrate the computational capabilities of our low-rank method, we compare the EM-based estimation approach with Bayesian inference via SPDE-INLA. The results demonstrate that the EM-based approach achieves substantial computational gains relative to SPDE-INLA, particularly as the number of time steps increases, while maintaining comparable predictive accuracy. This comparison should be interpreted as illustrative rather than definitive; its primary purpose is to show that the proposed MLE-based approach yields stable and statistically well-behaved estimates while remaining computationally competitive.

While Bayesian approaches offer the advantage of naturally deriving parameter standard deviations, they are susceptible to bias resulting from misspecified priors. Conversely, frequentist approaches avoid this bias but traditionally require manually differentiating the observed likelihood to estimate standard errors, a process that can complicate the inference procedure. We leverage vectorisation to scale the computation using the JAX Python package (Bradbury et al., 2018). JAX is a library for high-performance computing that combines a NumPy-like API with automatic differentiation and just-in-time compilation via XLA. The automatic differentiation addresses the common bottleneck in statistical inference by efficiently and automatically computing the likelihood derivatives with respect to the parameters. Moreover, the current implementation scales across all available local CPU cores and is designed for GPU/TPU execution.

The architectural choice of using the SPDE-GMRF approach offers versatility and potential for extension. Because the methodology relies on a mesh and a sparse precision structure, the entire framework can be straightforwardly transferred from planar regions to curved two-dimensional manifolds, such as the sphere, or adapted to discrete networks, such as road systems, power grids, or social graphs (Krainski et al., 2018), by simply modifying the underlying geometry or basis functions. In all these extensions, the preservation of the sparse precision matrix ensures that the core linear-time Kalman–EM machinery remains intact.

Supplementary materials

A supplement to the main article, including additional simulated results, is provided in the supplementary materials. (.pdf file).

The Python code (.py script) and instructions for reproducing the results in the article are available at the GitHub repository: https://github.com/jacopoRodeschini/Low_Rank_State_Space_Model.

Declaration of competing interest

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

Acknowledgements

This work is partially funded by the National Recovery and Resilience Plan, Italy (NRRP), Mission 4 Component 2 Investment 1.3, under Call for tender No. 341 of 15/03/2022, funded by the Italian Ministry of University and Research and the European Union's NextGenerationEU initiative. Award Number: PE 00000018, Concession Decree No. 1558 of 11/10/2022, CUP F83C22001720001, under the project titled Growing Resilient Inclusive and Sustainable (GRINS). Lorenzo Tedesco is also a member of the GNCS group of INdAM, whose support is gratefully acknowledged. This research was partially supported by the German Research Foundation, project number 501539976 (Philipp Otto).

Appendix

This appendix collects supporting material for the main text. Appendix A defines the notation used throughout the appendix. Appendix B provides the detailed proof of the identifiability theorem. Appendix C discusses the weak convergence properties of the low-rank approximation to the latent process, with full theoretical justifications. Appendix E presents the derivation and implementation details of the EM algorithm used for inference.

Appendix A. Notation

We introduce the following notation. For two matrices A and B of dimensions $r_A \times c$ and $r_B \times c$, respectively, we denote by $[A; B]$ the $(r_A + r_B) \times c$ matrix obtained by stacking A on top of B. We use $A \otimes B$ to denote the Kronecker product. Analogously, for two matrices A and B with dimensions $r \times c_A$ and $r \times c_B$, respectively, we define the column-wise juxtaposition of A and B, denoted by $[A, B]$, as the $r \times (c_A + c_B)$ matrix obtained by placing A to the left of B. Moreover, for matrices A and B of dimensions $r_A \times c_A$ and $r_B \times c_B$, respectively, we denote by $A \oplus B$ the block diagonal matrix of dimension $(r_A + r_B) \times (c_A + c_B)$, constructed with A and B placed as diagonal blocks and zeros elsewhere. Lastly, denote by $|A|$ the determinant of a square matrix A and by $\boldsymbol{I}_n$ the identity matrix of dimension n.

Appendix B. Proof of Theorem 1

Proof. Let

$$\Pi = (\boldsymbol{\beta}, \sigma^2, \boldsymbol{f}, \boldsymbol{W}, \{k_i\}_{i=1}^q), \qquad \Pi^\star = (\boldsymbol{\beta}^\star, \sigma^{2\star}, \boldsymbol{f}^\star, \boldsymbol{W}^\star, \{k_i^\star\}_{i=1}^q),$$

be two parameter sets that yield the same joint distribution for the observable field $\{\boldsymbol{y}(\boldsymbol{s},t) : (\boldsymbol{s},t) \in \mathcal{S}\times\mathbb{N}\}$ under the LR-SSM (6)–(7) for a fixed R. We show leading to $\Pi = \Pi^\star$.

Premultiply the measurement equation by $\boldsymbol{X}(\boldsymbol{s},t)$ and integrate over $\mathcal{S}$:

$$\int_{\mathcal{S}} \boldsymbol{X}(\boldsymbol{s},t)\boldsymbol{y}(\boldsymbol{s},t)\,d\boldsymbol{s} = \int_{\mathcal{S}} \boldsymbol{X}(\boldsymbol{s},t)\boldsymbol{X}(\boldsymbol{s},t)'\,d\boldsymbol{s}\,\boldsymbol{\beta} + \int_{\mathcal{S}} \boldsymbol{X}(\boldsymbol{s},t)\boldsymbol{W}\boldsymbol{\Psi}_R(\boldsymbol{s})\boldsymbol{z}_t\,d\boldsymbol{s} + \int_{\mathcal{S}} \boldsymbol{X}(\boldsymbol{s},t)\boldsymbol{\varepsilon}(\boldsymbol{s},t)\,d\boldsymbol{s}.$$

Average over time and define

$$\boldsymbol{g}_T = \frac{1}{T}\sum_{t=1}^{T}\int_{\mathcal{S}} \boldsymbol{X}(\boldsymbol{s},t)\boldsymbol{y}(\boldsymbol{s},t)\,d\boldsymbol{s}, \qquad \boldsymbol{L}_{X,T} = \frac{1}{T}\sum_{t=1}^{T}\int_{\mathcal{S}} \boldsymbol{X}(\boldsymbol{s},t)\boldsymbol{X}(\boldsymbol{s},t)'\,d\boldsymbol{s}.$$

Then

$$\boldsymbol{g}_T = \boldsymbol{L}_{X,T}\boldsymbol{\beta} + \boldsymbol{u}_T^{(z)} + \boldsymbol{u}_T^{(\varepsilon)},$$

where

$$\boldsymbol{u}_T^{(z)} = \frac{1}{T}\sum_{t=1}^{T}\int_{\mathcal{S}} \boldsymbol{X}(\boldsymbol{s},t)\boldsymbol{W}\boldsymbol{\Psi}_R(\boldsymbol{s})\boldsymbol{z}_t\,d\boldsymbol{s}, \qquad \boldsymbol{u}_T^{(\varepsilon)} = \frac{1}{T}\sum_{t=1}^{T}\int_{\mathcal{S}} \boldsymbol{X}(\boldsymbol{s},t)\boldsymbol{\varepsilon}(\boldsymbol{s},t)\,d\boldsymbol{s}.$$

By strict stationarity and ergodicity of $\{\boldsymbol{z}_t\}$ and the mean-orthogonality/exogeneity condition $\mathbb{E}[\boldsymbol{X}(\boldsymbol{s},t)\boldsymbol{W}\boldsymbol{\Psi}_R(\boldsymbol{s})\boldsymbol{z}_t] = \boldsymbol{0}$ (Assumption 1(iii)), together with boundedness and dominated convergence (and Fubini), we have $\boldsymbol{u}_T^{(z)} \to \boldsymbol{0}$ almost surely. Similarly, by the i.i.d. assumption on $\boldsymbol{\varepsilon}$ with $\mathbb{E}[\boldsymbol{X}(\boldsymbol{s},t)\boldsymbol{\varepsilon}(\boldsymbol{s},t)] = \boldsymbol{0}$ and dominated convergence, $\boldsymbol{u}_T^{(\varepsilon)} \to \boldsymbol{0}$ almost surely. Hence,

$$\boldsymbol{g}_T = \boldsymbol{L}_{X,T}\boldsymbol{\beta} + o_{\text{a.s.}}(1).$$

By Assumption 1(i), $\boldsymbol{L}_{X,T} \to \boldsymbol{L}_X$ and $\boldsymbol{g}_T \to \boldsymbol{g}$ (finite), where

$$\boldsymbol{L}_X = \lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\int_{\mathcal{S}} \boldsymbol{X}(\boldsymbol{s},t)\boldsymbol{X}(\boldsymbol{s},t)'\,d\boldsymbol{s}, \qquad \boldsymbol{g} = \lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\int_{\mathcal{S}} \boldsymbol{X}(\boldsymbol{s},t)\boldsymbol{y}(\boldsymbol{s},t)\,d\boldsymbol{s}.$$

Therefore $\boldsymbol{g} = \boldsymbol{L}_X\boldsymbol{\beta}$. Since $\boldsymbol{L}_X$ is nonsingular by Assumption 1(ii),

$$\boldsymbol{\beta} = \boldsymbol{L}_X^{-1}\boldsymbol{g},$$

so $\boldsymbol{\beta}$ is uniquely identified from the distribution of $\{\boldsymbol{y}(\boldsymbol{s},t)\}$. In particular, equality in distribution under Π and $\Pi^\star$ implies $\boldsymbol{\beta} = \boldsymbol{\beta}^\star$.

Define now the residual process

$$\boldsymbol{r}(\boldsymbol{s},t) = \boldsymbol{y}(\boldsymbol{s},t) - \boldsymbol{X}'(\boldsymbol{s},t)\boldsymbol{\beta} = \boldsymbol{W}\boldsymbol{\Psi}_R(\boldsymbol{s})\boldsymbol{z}_t + \boldsymbol{\varepsilon}(\boldsymbol{s},t),$$

and similarly $\boldsymbol{r}^\star(\boldsymbol{s},t) = \boldsymbol{y}(\boldsymbol{s},t) - \boldsymbol{X}'(\boldsymbol{s},t)\boldsymbol{\beta}^\star$. Since $\boldsymbol{\beta} = \boldsymbol{\beta}^\star$, $\boldsymbol{r}(\boldsymbol{s},t)$ and $\boldsymbol{r}^\star(\boldsymbol{s},t)$ coincide in distribution.

Let the lag-ℓ covariance kernel of $\boldsymbol{r}$ be

$$\boldsymbol{\Gamma}_\ell(\boldsymbol{s},\boldsymbol{s}') = \operatorname{Cov}\big(\boldsymbol{r}(\boldsymbol{s},t), \boldsymbol{r}(\boldsymbol{s}',t-\ell)\big), \qquad \ell = 0,1.$$

Using temporal whiteness of $\boldsymbol{\varepsilon}$, we have for $\ell = 1$,

$$\boldsymbol{\Gamma}_1(\boldsymbol{s},\boldsymbol{s}') = \boldsymbol{W}\boldsymbol{\Psi}_R(\boldsymbol{s})\,\operatorname{Cov}(\boldsymbol{z}_t,\boldsymbol{z}_{t-1})\,\boldsymbol{\Psi}_R(\boldsymbol{s}')'\boldsymbol{W}'.$$

Write $\boldsymbol{A} = \operatorname{diag}(\boldsymbol{f}) \otimes \boldsymbol{I}_R$. Stationarity implies

$$\operatorname{Cov}(\boldsymbol{z}_t,\boldsymbol{z}_{t-1}) = \boldsymbol{A}\boldsymbol{V}_z, \qquad \boldsymbol{V}_z = \operatorname{Var}(\boldsymbol{z}_t).$$

Let $\boldsymbol{W}^+$ be a left inverse of $\boldsymbol{W}$ (exists since $\boldsymbol{W}$ has full column rank by Assumption 2(ii)), so that $\boldsymbol{W}^+\boldsymbol{W} = \boldsymbol{I}_q$. Define

$$\boldsymbol{S}_\ell = \int_{\mathcal{S}}\int_{\mathcal{S}} \boldsymbol{\Psi}_R(\boldsymbol{s})'\boldsymbol{W}^+\boldsymbol{\Gamma}_\ell(\boldsymbol{s},\boldsymbol{s}')\boldsymbol{W}^{+\prime}\boldsymbol{\Psi}_R(\boldsymbol{s}')\,d\boldsymbol{s}\,d\boldsymbol{s}', \qquad \ell = 0,1.$$

Then, in the population limit,

$$\boldsymbol{S}_1 = \boldsymbol{A}\boldsymbol{G}, \qquad \boldsymbol{S}_0 = \boldsymbol{G},$$

where

$$\boldsymbol{G} = \int_{\mathcal{S}}\int_{\mathcal{S}} \boldsymbol{\Psi}_R(\boldsymbol{s})'\boldsymbol{V}_z\boldsymbol{\Psi}_R(\boldsymbol{s}')\,d\boldsymbol{s}\,d\boldsymbol{s}'.$$

By Assumption 2(iii) and the fact that $\boldsymbol{V}_z \succ 0$ under stationarity, $\boldsymbol{G}$ is positive definite. Hence $\boldsymbol{S}_0$ is invertible and

$$\boldsymbol{H} = \boldsymbol{S}_0^{-1}\boldsymbol{S}_1 = \boldsymbol{A} = \operatorname{diag}(\boldsymbol{f}) \otimes \boldsymbol{I}_R.$$

The eigenvalues of $\boldsymbol{H}$ are $\{f_1, \dots, f_q\}$, each with multiplicity R. Equality in distribution of observables under Π and $\Pi^\star$ implies $\boldsymbol{H} = \boldsymbol{H}^\star$, hence the multisets of eigenvalues coincide. Assumption 2(i) (strict ordering) fixes the correspondence, yielding $\boldsymbol{f} = \boldsymbol{f}^\star$.

Stack observations over a spatially dense set of locations $\{\boldsymbol{s}_j\}_{j=1}^m \subset \mathcal{S}$ at time t into $\boldsymbol{r}_t = (\boldsymbol{r}(\boldsymbol{s}_1, t)', \dots, \boldsymbol{r}(\boldsymbol{s}_m, t)')' \in \mathbb{R}^{mp}$. Let $\boldsymbol{P}$ denote the matrix collecting basis evaluations so that the stacked factor representation can be written as

$$\boldsymbol{r}_t = (\boldsymbol{I}_m \otimes \boldsymbol{W})\boldsymbol{P}\,\boldsymbol{z}_t + \boldsymbol{\epsilon}_t, \qquad \operatorname{Var}(\boldsymbol{\epsilon}_t) = \boldsymbol{I}_m \otimes \boldsymbol{\Sigma},$$

with $\boldsymbol{\Sigma} = \operatorname{diag}(\boldsymbol{\sigma}^2)$. Then

$$\operatorname{Cov}(\boldsymbol{r}_t) = (\boldsymbol{I}_m \otimes \boldsymbol{W})(\boldsymbol{P}\boldsymbol{V}_z\boldsymbol{P}')(\boldsymbol{I}_m \otimes \boldsymbol{W})' + \boldsymbol{I}_m \otimes \boldsymbol{\Sigma}.$$

Assumption 2(ii) gives full column rank of $\boldsymbol{W}$, and Assumption 2(iii) gives that the stacked evaluation matrix $\boldsymbol{P}$ has full column rank over a dense set of locations. Together with the column-scaling/sign convention in Assumption 2(ii), standard factor-analysis identification arguments (e.g., Bai and Wang, 2015) imply that $\boldsymbol{W}$ and $\boldsymbol{V}_z$ are uniquely identified, hence $\boldsymbol{W} = \boldsymbol{W}^\star$ and $\boldsymbol{V}_z = \boldsymbol{V}_z^\star$.

Under the LR-SPDE construction, $\boldsymbol{V}_z$ is block diagonal with blocks that are functions of the SPDE precision matrices $\boldsymbol{Q}_{k_i}$ (equivalently of k_i) and of $\boldsymbol{f}$, with $\boldsymbol{f}$ already identified. By Assumption 2(iv), the mapping $k \mapsto \boldsymbol{Q}_k$ is injective, so equality $\boldsymbol{V}_z = \boldsymbol{V}_z^\star$ implies $k_i = k_i^\star$ for all $i = 1, \dots, q$.

Finally, using the pointwise variance identity

$$\operatorname{Var}\big(\boldsymbol{r}(\boldsymbol{s}, t)\big) = \boldsymbol{W}\boldsymbol{\Psi}_R(\boldsymbol{s})\boldsymbol{V}_z\boldsymbol{\Psi}_R(\boldsymbol{s})'\boldsymbol{W}' + \boldsymbol{\Sigma},$$

and the fact that $\boldsymbol{W}$, $\boldsymbol{V}_z$, and $\boldsymbol{\Psi}_R(\boldsymbol{s})$ are identified, the diagonal matrix $\boldsymbol{\Sigma} = \operatorname{diag}(\boldsymbol{\sigma}^2)$ is uniquely determined from the marginal variances of $\boldsymbol{r}(\boldsymbol{s}, t)$. Hence $\boldsymbol{\sigma}^2 = \boldsymbol{\sigma}^{2\star}$.

Combining Steps 1–4 yields

$$\boldsymbol{\beta} = \boldsymbol{\beta}^\star, \quad \boldsymbol{f} = \boldsymbol{f}^\star, \quad \boldsymbol{W} = \boldsymbol{W}^\star, \quad k_i = k_i^\star \ \forall i, \quad \boldsymbol{\sigma}^2 = \boldsymbol{\sigma}^{2\star},$$

and therefore $\Pi = \Pi^\star$. Thus the parameter set of the LR-SSM is point-identified in distribution. □

Lemma 1. *Let $\boldsymbol{C}$ and $\boldsymbol{G}$ be the mass and stiffness matrices associated with a fixed finite element mesh, where $\boldsymbol{C}$ is symmetric positive definite and $\boldsymbol{G}$ is symmetric positive semidefinite. For $k \geq 0$ define*

$$\boldsymbol{Q}_k \;=\; (k^2\boldsymbol{C} + \boldsymbol{G})\,\boldsymbol{C}^{-1}\,(k^2\boldsymbol{C} + \boldsymbol{G}).$$

Then the map $k \mapsto \boldsymbol{Q}_k$ is injective: if $\boldsymbol{Q}_{k_1} = \boldsymbol{Q}_{k_2}$ for $k_1, k_2 \geq 0$, then $k_1 = k_2$. The same statement holds if $\boldsymbol{C}$ is replaced by a diagonal mass-matrix approximation $\widetilde{\boldsymbol{C}}$ that is positive definite.

Proof. Write $t = k^2 \geq 0$ and define $\boldsymbol{Q}(t) = \boldsymbol{Q}_k$. Expanding,

$$\boldsymbol{Q}(t) \;=\; \boldsymbol{C}\,t^2 + 2\boldsymbol{G}\,t + \boldsymbol{G}\,\boldsymbol{C}^{-1}\boldsymbol{G}.$$

Assume $\boldsymbol{Q}(t_1) = \boldsymbol{Q}(t_2)$ for some $t_1, t_2 \geq 0$. Subtracting gives

$$\boldsymbol{C}\,(t_1^2 - t_2^2) + 2\boldsymbol{G}\,(t_1 - t_2) \;=\; \boldsymbol{0},$$

hence

$$(t_1 - t_2)\,\big((t_1 + t_2)\boldsymbol{C} + 2\boldsymbol{G}\big) \;=\; \boldsymbol{0}.$$

If $t_1 \neq t_2$, then $(t_1 + t_2)\boldsymbol{C} + 2\boldsymbol{G} = \boldsymbol{0}$. But $(t_1 + t_2) \geq 0$, $\boldsymbol{C} \succ 0$, and $\boldsymbol{G} \succeq 0$, so $(t_1 + t_2)\boldsymbol{C} + 2\boldsymbol{G} \succeq (t_1 + t_2)\boldsymbol{C}$. If $t_1 + t_2 > 0$, this matrix is positive definite and cannot be the zero matrix, a contradiction. The remaining possibility $t_1 + t_2 = 0$ forces $t_1 = t_2 = 0$, contradicting $t_1 \neq t_2$. Therefore $t_1 = t_2$, and hence $k_1^2 = k_2^2$ with $k_1, k_2 \geq 0$ implies $k_1 = k_2$.

If $\boldsymbol{C}$ is replaced by a diagonal approximation $\widetilde{\boldsymbol{C}} \succ 0$, the same algebra shows that

$$(t_1 - t_2)\,\big((t_1 + t_2)\widetilde{\boldsymbol{C}} + 2\boldsymbol{G}\big) \;=\; \boldsymbol{0},$$

and the identical positive-(semi)definiteness argument yields $t_1 = t_2$. □

Appendix C. Weak convergence of the low-rank approximation

In this section, we discuss the weak convergence of the process $\boldsymbol{\Psi}_R(\boldsymbol{s})\boldsymbol{z}(\mathcal{V}_R, t)$ to the continuous process $\boldsymbol{z}(\boldsymbol{s}, t)$. To this end, we first introduce the concept of Gaussian Markov Random Fields (GMRFs).

Let $G = (\mathcal{V}, \mathcal{E})$ be a finite undirected graph, where $\mathcal{V}$ is a set of vertices with cardinality $R < \infty$, and $\mathcal{E}$ is the set of edges. Let $\mathcal{X} = \{\mathcal{X}_v\}_{v \in \mathcal{V}}$ denote a collection of random variables indexed by $\mathcal{V}$. We say that $\mathcal{X}$ forms a zero-mean Markov Random Field defined by G if it satisfies the local Markov property:

$$\mathcal{X}_i \perp\!\!\!\perp \mathcal{X}_{\mathcal{V}\setminus\partial i} \mid \mathcal{X}_{\partial i}, \quad \forall i \in \mathcal{V}, \tag{11}$$

where ∂i denotes the set of neighbours of node i, i.e., all vertices adjacent to i in the graph. Specifically, $\mathcal{X}$ is a GMRF on G if it is Gaussian with zero mean and precision matrix Q such that $Q_{ij} = 0$ whenever $j \notin \{\partial i \cup i\}$. This sparsity in the precision matrix facilitates the use of efficient numerical methods.

The explicit link between Gaussian Field (GF) parameters and GMRF parameters is established in Lindgren et al. (2011). The goal is to approximate a GF with covariance matrix Σ using a GMRF with precision matrix Q such that Q^{-1} approximates Σ in a suitable norm. A GMRF representation of a Matérn GF can be efficiently constructed using the stochastic partial differential equation (SPDE) approach, in combination with the Finite Element Method (FEM) over a triangulated domain.

Let $\Omega \subset \mathbb{R}^d$ be a bounded domain with a smooth boundary $\partial\Omega \in C^2$. As discussed in Whittle (1954), Lindgren et al. (2011), a Gaussian field $x(\boldsymbol{s})$ on $\mathbb{R}^d$ with Matérn covariance characterised by parameters (σ^2, ν, κ) is a stationary solution of the SPDE:

$$(\kappa^2 - \Delta)^{\alpha/2} x(\boldsymbol{s}) = \mathcal{W}(\boldsymbol{s}), \quad \boldsymbol{s} \in \mathbb{R}^d, \quad \kappa > 0, \tag{12}$$

where $\Delta = \sum_{i=1}^{d} \frac{\partial^2}{\partial x_i^2}$ is the Laplacian and $\alpha = \nu + d/2$. The operator $(\kappa^2 - \Delta)^{\alpha/2}$ is defined via its spectral properties, as described in Whittle (1954, 1963). Here, $\mathcal{W}$ denotes spatial Gaussian white noise with unit variance, and $x(\boldsymbol{s})$ has marginal variance:

$$\sigma^2 = \frac{\Gamma(\nu)}{\Gamma(\nu + d/2)(4\pi)^{d/2}\kappa^{2\nu}}.$$

For simplicity, we consider the common spatial case with $d = 2$ and $\alpha = 2$, implying $\sigma^2 = (4\pi\kappa^2)^{-1}$. The spectral density of the stationary solution to (12) is given by:

$$R(\boldsymbol{k}) = (2\pi)^{-2}(\kappa^2 + \|\boldsymbol{k}\|^2)^{-2}.$$

Using the Fourier transform, the fractional Laplacian is defined as:

$$\mathcal{F}\{(\kappa^2 - \Delta)\varphi\}(\boldsymbol{k}) = (\kappa^2 + \|\boldsymbol{k}\|^2)\mathcal{F}\varphi(\boldsymbol{k}), \tag{13}$$

where φ is any function on $\mathbb{R}^d$ with an invertible right-hand side under the Fourier transform. To avoid null-space solutions, Neumann boundary conditions (zero normal derivatives on $\partial\Omega$) are imposed.

To numerically solve (12) restricted on Ω, we apply FEM (Ciarlet, 2002; Brenner, 2008; Quarteroni and Valli, 2008). Let Ω be partitioned into non-overlapping triangles via Delaunay triangulation, resulting in a mesh graph $G = (\mathcal{V}, \mathcal{E})$ with $R = |\mathcal{V}|$.

The stochastic weak formulation of the SPDE requires defining the inner product:

$$\langle f, g \rangle = \int_{\Omega} f(\boldsymbol{s}) g(\boldsymbol{s})\, d\boldsymbol{s}.$$

The weak solution satisfies:

$$\langle \phi_j, (\kappa^2 - \Delta)x \rangle \overset{w}{=} \langle \phi_j, \mathcal{W} \rangle, \quad \forall j, \tag{14}$$

for suitable test functions $\{\phi_j\}_{j=1}^{R}$. We represent the FEM approximation as:

$$x^R(\boldsymbol{s}) = \sum_{k=1}^{R} \psi_k(\boldsymbol{s}) w_k = \boldsymbol{\psi}_R' \boldsymbol{w}, \tag{15}$$

where $\boldsymbol{\psi}_R(\boldsymbol{s}) = (\psi_1(\boldsymbol{s}), \ldots, \psi_R(\boldsymbol{s}))'$ are piecewise linear basis functions, and $\boldsymbol{w}$ are Gaussian weights. Boundary conditions are enforced by assuming that the solution satisfies a Neumann condition on $\partial\Omega$, which in the finite element formulation is imposed weakly through the variational form. This ensures that the resulting approximation resides in the appropriate Hilbert space $\mathcal{H}_R(\Omega)$.

Choosing $\phi_k = \psi_k$ yields the Galerkin approximation. Define matrices:

$$\begin{aligned} \boldsymbol{C}_{ij} &= \langle \psi_i, \psi_j \rangle, \\ \boldsymbol{G}_{ij} &= \langle \nabla\psi_i, \nabla\psi_j \rangle, \\ \boldsymbol{K}_{\kappa^2} &= \kappa^2 \boldsymbol{C} + \boldsymbol{G}. \end{aligned}$$

Under Neumann conditions, the precision matrix for $\boldsymbol{w}$ becomes:

$$\boldsymbol{Q}_{\kappa^2} = \boldsymbol{K}_{\kappa^2} \boldsymbol{C}^{-1} \boldsymbol{K}_{\kappa^2}. \tag{16}$$

While $\boldsymbol{C}^{-1}$ is dense, one can replace $\boldsymbol{C}$ with a diagonal matrix $\tilde{\boldsymbol{C}}$, where $\tilde{\boldsymbol{C}}_{ii} = \langle \psi_i, 1 \rangle$, to obtain a sparse $\boldsymbol{Q}$ (Lindgren et al., 2011).

As shown in equation (11) of Lindgren et al. (2011), the approximation error is bounded as follows:

$$\sup_{f \in \mathcal{H}^1, \|f\|_{\mathcal{H}^1} \leq 1} \mathbb{E}\left[\langle f, x^R - x \rangle_{\mathcal{H}^1}^2 \right] \leq c h^2, \tag{17}$$

where $\mathcal{H}^1(\Omega, \kappa)$, endowed with the internal product

$$\langle \phi, \psi \rangle_{\mathcal{H}^1(\Omega,\kappa)} = \kappa^2 \langle \phi, \psi \rangle + \langle \nabla\phi, \nabla\psi \rangle,$$

is the Hilbert space, and h denotes the mesh resolution.

We consider the following convergence definition,as formulated in Lindgren et al. (2011). Let x^R be a sequence of $L^2(\Omega)$-bounded Gaussian fields. We say $x^R \xrightarrow{D\{L^2(\Omega)\}} x$ if for all $f, g \in L^2(\Omega)$,

$$\mathbb{E}[\langle f, x^R\rangle] \to \mathbb{E}[\langle f, x\rangle], \quad \text{and} \quad \mathrm{Cov}(\langle f, x^R\rangle, \langle g, x^R\rangle) \to \mathrm{Cov}(\langle f, x\rangle, \langle g, x\rangle).$$

Theorem 4. *Let x be a weak solution to the SPDE $\mathcal{L}x = \mathcal{W}$ with $\mathcal{L} = (\kappa^2 - \Delta)$ and Neumann boundary conditions. Let x^R be its FEM approximation in $\mathcal{H}^1_R(\Omega, \kappa)$ using Gaussian white noise $\mathcal{W}$. Then,*

$$x^R \xrightarrow{D\{L^2(\Omega)\}} x,$$
$$\mathcal{L}x^R \xrightarrow{D\{L^2(\Omega)\}} \mathcal{L}x,$$

for $R \to \infty$ provided that $\bigcup_R \mathcal{H}^1_R(\Omega, \kappa)$ is dense in $\mathcal{H}^1(\Omega, \kappa)$ and x^R is the Galerkin solution.

The proof is available in Lindgren et al. (2011).

C.1. Proof of Corollary 1

Proof. By Theorem 4, and fixing the component i of the process $\boldsymbol{\eta}(\cdot, t)$ model, we have $\eta_i(\cdot, t)^R \xrightarrow{D\{L^2(\Omega)\}} \eta_i(\cdot, t)$ where $\eta_i^R(\cdot, t) = \boldsymbol{\psi}_R(s)'\boldsymbol{\eta}_i(\mathcal{V}_R, t)$ for some Gaussian weights $\boldsymbol{\eta}_i(\mathcal{V}_R, t)$, for each $t = 0, \dots, T$. Due to independence, and the fact that $t = 1, \dots, T$ varies over a finite set, we have that $\eta_i^R(\cdot, \cdot) \xrightarrow{D\{L^2(\Omega^T)\}} \eta_i(\cdot, \cdot)$. Now, due to SSM (6)–(7), we have that $z_i(\cdot, t) = \sum_{j=0}^t f_i^j \eta_i(\cdot, j)$. Thus, by the linearity of the approximation $\eta_i^R(\cdot, t)$, we have an immediate approximation of $z_i(\cdot, \cdot)$ as $z_i^R(\boldsymbol{s}, t) = \boldsymbol{\psi}_R(\boldsymbol{s})\boldsymbol{z}_i(\mathcal{V}_R, t)$ where the Gaussian weights are $\boldsymbol{z}_i(\mathcal{V}_R, t) = \sum_{j=0}^T f_i^{t-j}\boldsymbol{\eta}_i(\mathcal{V}_R, t)$. Therefore, $\eta_i^R(\cdot, \cdot) \xrightarrow{D\{L^2(\Omega^T)\}} \eta_i(\cdot, \cdot)$ implies $z_i^R(\cdot, \cdot) \xrightarrow{D\{L^2(\Omega^T)\}} z_i(\cdot, \cdot)$. As the number of components is finite $i = 1, \dots, q$, we obtain the aimed result:

$$\boldsymbol{z}^R(\boldsymbol{s}, t) = \boldsymbol{\Psi}_R(\boldsymbol{s})\boldsymbol{z}(\mathcal{V}_R, t) \xrightarrow{D\{L^2(\Omega^{qT})\}} \boldsymbol{z}(\boldsymbol{s}, t).$$

This convergence result also implies that

$$\mathbb{E}[\boldsymbol{\Psi}_R(\boldsymbol{s})\boldsymbol{z}(\mathcal{V}_R, t)] \to \mathbb{E}[\boldsymbol{z}(\boldsymbol{s}, t)],$$
$$\mathrm{Cov}(\boldsymbol{\Psi}_R(\boldsymbol{s})\boldsymbol{z}(\mathcal{V}_R, t), \boldsymbol{\Psi}_R(\tilde{\boldsymbol{s}})\boldsymbol{z}(\mathcal{V}_R, \tilde{t})) \to \mathrm{Cov}(\boldsymbol{z}(\boldsymbol{s}, t), \boldsymbol{z}(\tilde{\boldsymbol{s}}, \tilde{t})).$$

□ □

C.2. Proof of Theorem 2

Proof. The two observation Eqs. (1) and (6) differ only in the latent state, therefore

$$\begin{aligned}\boldsymbol{y}(\boldsymbol{s}, t) - \boldsymbol{y}^R(\boldsymbol{s}, t) &= \boldsymbol{X}_t'\boldsymbol{\beta} + \boldsymbol{W}\boldsymbol{z}(\boldsymbol{s}, t) + \boldsymbol{\varepsilon}(\boldsymbol{s}, t) - \boldsymbol{X}_t'\boldsymbol{\beta} + \boldsymbol{W}\boldsymbol{z}^R(\boldsymbol{s}, t) + \boldsymbol{\varepsilon}(\boldsymbol{s}, t)\\ &= \boldsymbol{W}(\boldsymbol{z}(\boldsymbol{s}, t) - \boldsymbol{z}^R(\boldsymbol{s}, t)),\end{aligned}$$

so the mean-square bias $\mathbb{E}\left[\|\boldsymbol{y}_t - \boldsymbol{y}_t^R\|^2\right]$ is controlled by the mean-square error of the state approximation:

$$\mathbb{E}\left[\|\boldsymbol{y}_t - \boldsymbol{y}_t^R\|_2^2\right] \le C \cdot \mathbb{E}\left[\|\boldsymbol{z}_t - \boldsymbol{z}_t^R\|_2^2\right], \tag{18}$$

where C is a constant that depends on $\|W\|$. The inequality (17) implies

$$\mathbb{E}\left[\|\eta_i^R - \eta_i\|_{\mathcal{H}^{-1}}^2\right] \le c\, h^2,$$

for each component $i = 1, \dots, q$ of $\boldsymbol{\eta}$ where $\|\cdot\|_{\mathcal{H}^{-1}}$ is the dual norm of $\mathcal{H}^1$. On the bounded domain Ω, the inclusion $L^2(\Omega) \subset \mathcal{H}^{-1}(\Omega)$ leads to

$$\|g\|_{L^2} \le C_{\mathrm{emb}}\, \|g\|_{H^{-1}} \quad \text{(for every } g \in H^{-1} \text{ with zero mean)},$$

that applied to $g = \boldsymbol{\eta}_t^R - \boldsymbol{\eta}_t$ and taking expectation leads to

$$\mathbb{E}\left[\|\boldsymbol{\eta}^R - \boldsymbol{\eta}\|_{L^2}^2\right] \le C_{\mathrm{emb}}^2 \sum_{i=1}^q \mathbb{E}\left[\|\eta_i^R - \eta_i\|_{H^{-1}}^2\right] \le q\, C_{\mathrm{emb}}^2\, c\, h^2.$$

As we are interested in a general time $t = 1, \dots, T$, the AR(1) Eqs. (2) and (7) yield

$$\begin{aligned}\|\boldsymbol{z}_t - \boldsymbol{z}_t^R\|_2 &\le \varphi\|\boldsymbol{z}_{t-1} - \boldsymbol{z}_{t-1}^R\|_2 + \|\boldsymbol{\eta}_t - \boldsymbol{\eta}_t^R\|_2\\ &\le \varphi\|\boldsymbol{z}_{t-1} - \boldsymbol{z}_{t-1}^R\|_2 + q\, C_{\mathrm{emb}}^2\, c\, h^2,\end{aligned}$$

with C being a constant $\varphi = \max_i |f_i| < 1$. Iterating backwards in time, we obtain

$$\mathbb{E}\left[\|\boldsymbol{z}_t^R - \boldsymbol{z}_t\|_{L^2}^2\right] \le \sum_{r=0}^t \varphi^{2r}\, q\, C_{\mathrm{emb}}^2\, c\, h^2 = \frac{1 - \varphi^{2(t+1)}}{1 - \varphi^2}\, q\, C_{\mathrm{emb}}^2\, c\, h^2, \tag{19}$$

Eqs. (18) and (19) yield the result. □ □

Appendix D. Proof of Theorem 3

Let $\boldsymbol{y}(\boldsymbol{s},t)$ distributed as specified in SSM (1)–(2) with set of parameter Π. We are interested in studying the bias on the estimation of parameters when, for a fixed R we use the LR-SSM in the estimation. Let $\boldsymbol{\pi}$ the vector $(\boldsymbol{\beta}', \boldsymbol{\sigma}', \boldsymbol{f}', k_1, \dots, k_q)'$. Denote by

$$\mathcal{P}_{\boldsymbol{\pi}} \equiv \mathcal{N}\big(\boldsymbol{X}'(\boldsymbol{s},t)\boldsymbol{\beta}, \boldsymbol{\Sigma}_y(\boldsymbol{\pi})\big), \qquad \boldsymbol{\Sigma}_y(\boldsymbol{\pi}) = \boldsymbol{W}\boldsymbol{\Sigma}_z(\boldsymbol{\pi})\boldsymbol{W}' + \boldsymbol{\Sigma},$$

the law of $\boldsymbol{y}(\boldsymbol{s},t)$, with $\boldsymbol{\Sigma} = \operatorname{diag}(\boldsymbol{\sigma}^2)$ and latent covariance $\boldsymbol{\Sigma}_z(\boldsymbol{\pi})$. Replacing $\boldsymbol{\eta}(\boldsymbol{s},t)$ by its FEM-approximation $\boldsymbol{\Psi}_R(\boldsymbol{s})\boldsymbol{\eta}(\mathcal{V}_R,t)$ yields a low-rank covariance

$$\boldsymbol{\Sigma}_y^{(R)}(\boldsymbol{\pi}) = \boldsymbol{W}\boldsymbol{\Sigma}_{z,R}(\boldsymbol{\pi})\boldsymbol{W}' + \boldsymbol{\Sigma}.$$

Define the covariance perturbation

$$\boldsymbol{\Delta}_R(\boldsymbol{\pi}) = \boldsymbol{\Sigma}_z(\boldsymbol{\pi}) - \boldsymbol{\Sigma}_{z,R}(\boldsymbol{\pi}).$$

By the FEM approximation, see equation (11) of Lindgren et al. (2011), the Galerkin error is bounded as

$$\left\|\Delta_R(\boldsymbol{\pi})\right\| \le C\,h, \tag{20}$$

where h is the diameter of the largest circle that can be inscribed in a triangle in the triangulation, and C is a constant. Because the LR-SSM is misspecified, the likelihood based on (6)–(7) concentrates around the minimiser of the Kullback–Leibler divergence

$$\boldsymbol{\pi}_R^{\star} = \arg\min_{\boldsymbol{\vartheta}\in\Theta_R} \operatorname{KL}\big(\mathcal{P}_{\boldsymbol{\pi}} \,\|\, \mathcal{P}_{\boldsymbol{\vartheta}}^{(R)}\big), \tag{21}$$

where $\mathcal{P}_{\boldsymbol{\vartheta}}^{(R)} \equiv \mathcal{N}(\boldsymbol{X}(\boldsymbol{s},t)'\boldsymbol{\beta}, \boldsymbol{\Sigma}_y^{(R)}(\boldsymbol{\vartheta}))$. For Gaussian measures with the same mean function, the Kullback–Leibler divergence takes the form

$$\operatorname{KL}(\boldsymbol{\pi},\boldsymbol{\vartheta}) = \tfrac{1}{2}\left\{\operatorname{tr}\big[\boldsymbol{\Sigma}_y^{(R)}(\boldsymbol{\vartheta})^{-1}\boldsymbol{\Sigma}_y(\boldsymbol{\pi})\big] - \log\det\big[\boldsymbol{\Sigma}_y^{(R)}(\boldsymbol{\vartheta})^{-1}\boldsymbol{\Sigma}_y(\boldsymbol{\pi})\big] - p\right\}. \tag{22}$$

Setting the gradient to zero yields, after vectorising (vec) the matrices,

$$0 = \boldsymbol{J}'\boldsymbol{\Sigma}_y^{-1}(\boldsymbol{\pi}) \otimes \boldsymbol{\Sigma}_y^{-1}(\boldsymbol{\pi})\ \operatorname{vec}\boldsymbol{\Delta}_R(\boldsymbol{\pi}) + \ \boldsymbol{H}\,(\boldsymbol{\pi}_R^{\star} - \boldsymbol{\pi}) + o\big(\|\Delta_R(\boldsymbol{\pi})\|\big),$$

where $\boldsymbol{J} = \nabla_{\boldsymbol{\vartheta}} \operatorname{vec}\boldsymbol{\Sigma}_y^{(R)}(\boldsymbol{\pi})\big|_{\boldsymbol{\vartheta}=\boldsymbol{\pi}}$ and $\boldsymbol{H}$ is the Fisher-information matrix of the correct model, which is positive definite. Solving the equation, we obtain the exact first-order bias

$$\boldsymbol{\pi}_R^{\star} - \boldsymbol{\pi} = -\boldsymbol{H}^{-1}\boldsymbol{J}'\big(\boldsymbol{\Sigma}_y^{-1}(\boldsymbol{\pi}) \otimes \boldsymbol{\Sigma}_y^{-1}(\boldsymbol{\pi})\big)\operatorname{vec}\boldsymbol{\Delta}_R(\boldsymbol{\pi}) \ + \ o\big(\|\boldsymbol{\Delta}_R(\boldsymbol{\pi})\|\big). \tag{23}$$

Combining (23) with the covariance bound (20) delivers the rate

$$\left\|\boldsymbol{\pi}_R^{\star} - \boldsymbol{\pi}\right\| = \mathcal{O}(h).$$

Note that, the term (22) does not involve the mean of the processes, which implies the parameter $\boldsymbol{\beta}$ has no bias. □

Appendix E. EM algorithm

We are interested in estimating the set parameter Π based on the observed data. We recall here the notation. Denote by $\boldsymbol{y}_t$ the $(m_t = \sum_{j=1}^{p} m_{j,t})$-dimensional vector

$$\boldsymbol{y}_t = \big(\boldsymbol{y}_1(\mathcal{S}_{1,t},t)', \boldsymbol{y}_2(\mathcal{S}_{2,t},t)', \dots, \boldsymbol{y}_p(\mathcal{S}_{p,t},t)'\big)',$$

where $\mathcal{S}_{j,t} = \{\boldsymbol{s}_{i,t}\}_{i=1}^{m_{j,t}}$ is the set of locations where the process $y_j(\cdot,t)$ has been observed, and

$$\boldsymbol{y}_{j,t} = \boldsymbol{y}_j(\mathcal{S}_{j,t},t) = \Big(y_j(\boldsymbol{s}_{1,t},t), \dots, y_j(\boldsymbol{s}_{m_{j,t},t},t)\Big)'.$$

Let $\boldsymbol{y}_{1:T}$ denote the vector $(\boldsymbol{y}_1', \dots, \boldsymbol{y}_T')'$.

An analogous notation is used for $\boldsymbol{z}_{1:T} = (\boldsymbol{z}_1', \dots, \boldsymbol{z}_T')'$, where

$$\boldsymbol{z}_t = \big(z_1(\mathcal{V}_R,t)', \dots, \boldsymbol{z}_q(\mathcal{V}_R,t)'\big)', \quad \text{with} \quad \boldsymbol{z}_{j,t} = \boldsymbol{z}_j(\mathcal{V}_R,t) = \big(z_j(\boldsymbol{r}_1,t), \dots, z_j(\boldsymbol{r}_R,t)\big)'.$$

Moreover, let $\boldsymbol{X}_{1:T} = [\boldsymbol{X}_1; \dots; \boldsymbol{X}_T]$, where each $\boldsymbol{X}_t$ is a $b \times m_t$ matrix, with $b = \sum_{j=1}^{p} b_j$, defined as

$$\boldsymbol{X}_t = \boldsymbol{X}_{1,t} \oplus \boldsymbol{X}_{2,t} \oplus \cdots \oplus \boldsymbol{X}_{p,t}.$$

Each component matrix $\boldsymbol{X}_{j,t}$ is constructed as

$$\boldsymbol{X}_{j,t} = [\boldsymbol{X}_j(\boldsymbol{s}_{1,t},t), \dots, \boldsymbol{X}_j(\boldsymbol{s}_{m_{j,t},t},t)],$$

with $\boldsymbol{X}_{j,t}$ being a $b_j \times m_{j,t}$ matrix, and each column vector $\boldsymbol{X}_j(\boldsymbol{s},t) \in \mathbb{R}^{b_j}$ representing the covariate values associated with the jth component of $\boldsymbol{y}(\boldsymbol{s},t)$ at location $\boldsymbol{s}$ and time t.

We assume that the mesh $\mathcal{G}_R = (\mathcal{V}_R, \mathcal{E}_R)$, on which the latent process in the state–space model (6)–(7) is defined, is given. The procedure for selecting this mesh is discussed in Section 3.1.

Let $L(\Pi; \boldsymbol{y}_{1:T}, \boldsymbol{X}_{1:T}, \boldsymbol{z}_{1:T}) = p(\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T} | \boldsymbol{X}_{1:T}; \Pi)$ be the full likelihood function. The MLE of Π is obtained by maximising the marginal likelihood of the observed data likelihood

$$p(\boldsymbol{y}_{1:T} | \boldsymbol{X}_{1:T}; \Pi) = \int p(\boldsymbol{y}_{1:T} | \boldsymbol{X}_{1:T}, \boldsymbol{z}_{1:T}, \Pi) p(\boldsymbol{z}_{1:T} | \Pi) d\boldsymbol{z}_{1:T}. \tag{24}$$

However, this quantity is intractable since $\boldsymbol{z}_{1:T}$ is unobserved and its distribution is unknown before attaining Π. The EM algorithm seeks to find the ML estimate of the marginal likelihood by iteratively applying these two steps:

E-step Define $Q(\Pi \mid \Pi^{(k)})$ as the expected value of the log-likelihood function of Π, with respect to the current conditional distribution of $\boldsymbol{z}_{1:T}$ given $(\boldsymbol{y}_{1:T}, \boldsymbol{X}_{1:T})$ and the current estimates of the parameters $\Pi^{(k)}$:

$$Q(\Pi \mid \Pi^{(k)}) = \mathbb{E}_{\boldsymbol{z}_{1:T} \sim p(\cdot | \boldsymbol{y}_{1:T}, \boldsymbol{X}_{1:T}; \Pi^{(k)})} [\log p(\boldsymbol{y}_{1:T} \mid \boldsymbol{X}_{1:T}, \boldsymbol{z}_{1:T}; \Pi)].$$

M-step Obtain the next value of Π, namely $\Pi^{(k+1)}$ as the maximiser of $Q(\Pi | \Pi^{(k)})$:

$$\Pi^{(k+1)} = \arg\max_{\Pi \in \mathcal{P}} Q(\Pi \mid \Pi^{(k)}), \tag{25}$$

where the maximisation is done on a suitable compact set $\mathcal{P}$ which contains Π as interior point.

The computation of the conditional distribution of the latent states $\boldsymbol{z}_{1:T}$ given the observed data and the current value of the parameter $\Pi^{(i)}$ can be efficiently obtained using the Kalman filter and smoother, and the formulas are given in Appendix E together with the formulas to obtain arg max in (25). Several stopping criteria can be considered, including a maximum number of iterations, a tolerance on the norm of the difference $\Pi^{(k)} - \Pi^{(k+1)}$, or a tolerance on the percentage change in the likelihood. Suppose that the algorithm stops after K iterations, then the value $\hat{\Pi} = \Pi^{(K)}$ corresponds to the proposed estimator of Π.

Close-form formulas. We now obtain close-form formulas of the parameters $\{\boldsymbol{\beta}, \sigma^2, \boldsymbol{f}, \boldsymbol{w}\}$ and express the objective function that we need to numerically minimise to obtain an estimate of the value of κ_i for $i = 1, \dots, q$. First, we write explicitly the full log-likelihood $\ell(\Pi; \boldsymbol{y}_{1:T}, \boldsymbol{x}_{1:T}, \boldsymbol{z}_{1:T}) = \log p(\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T} | \boldsymbol{x}_{1:T}; \Pi)$. Denote by w_{ij} the element at row i and column j of the matrix $\boldsymbol{W}$ in (6) and define the matrix $\boldsymbol{\Psi}_t^{\boldsymbol{w}}$ as

$$\boldsymbol{\Psi}_t^{\boldsymbol{w}} = \begin{bmatrix} w_{11}\boldsymbol{\Psi}_{11,t} & \dots & w_{1q}\boldsymbol{\Psi}_{1q,t} \\ \vdots & \ddots & \vdots \\ w_{p1}\boldsymbol{\Psi}_{p1,t} & \dots & w_{pq}\boldsymbol{\Psi}_{pq,t} \end{bmatrix}, \tag{26}$$

where $\boldsymbol{\Psi}_{ij,t}$ is a known basis matrix that characterises the effect of the vector $\boldsymbol{z}_j(\mathcal{V}_j, t)$ on $\boldsymbol{y}_i(S_{i,t}, t)$, that is $\boldsymbol{\Psi}_{ij,t} = [\boldsymbol{\psi}_R(\boldsymbol{s}_{1,t})'; \dots; \boldsymbol{\psi}_R(\boldsymbol{s}_{m_{i,t},t})']$. Using the notation introduced in Section 3, we can write the SSM (6)–(7) in matrix form as

$$\begin{aligned} \boldsymbol{y}_t &= \boldsymbol{X}_t'\boldsymbol{\beta} + \boldsymbol{\Psi}_t^{\boldsymbol{w}}\boldsymbol{z}_t + \boldsymbol{\epsilon}_t, \\ \boldsymbol{z}_t &= \boldsymbol{F}\boldsymbol{z}_{t-1} + \boldsymbol{\eta}_t, \end{aligned}$$

where $\boldsymbol{\epsilon}_t = (\boldsymbol{\epsilon}_1(S_{1,t}, t)', \dots, \boldsymbol{\epsilon}_p(S_{p,t}, t))'$ with $\boldsymbol{\epsilon}_i(S_{i,t}, t) = (\epsilon_i(\boldsymbol{s}_{1,t}, t), \dots, \epsilon_i(\boldsymbol{s}_{m_{i,t},t}, t))'$. Thus, the full log-likelihood $\ell(\Pi; \boldsymbol{y}_{1:T}, \boldsymbol{X}_{1:T}, \boldsymbol{z}_{1:T})$ satisfies

$$\begin{aligned} -2\ell(\Pi; \boldsymbol{y}_{1:T}, \boldsymbol{X}_{1:T}, \boldsymbol{z}_{1:T}) &= \log|\boldsymbol{\Sigma}_0| + (\boldsymbol{z}_0 - \boldsymbol{\mu}_0)'\boldsymbol{\Sigma}_0^{-1}(\boldsymbol{z}_0 - \boldsymbol{\mu}_0) \\ &+ \sum_{t=1}^{T} \left(\log|\boldsymbol{\Sigma}_t| + \boldsymbol{e}_t'\boldsymbol{\Sigma}_t^{-1}\boldsymbol{e}_t\right) - T\log|\boldsymbol{Q}_{\boldsymbol{\kappa}}| + \sum_{t=1}^{T} \left(\boldsymbol{z}_t - \boldsymbol{F}\boldsymbol{z}_{t-1}\right)' \boldsymbol{Q}_{\boldsymbol{\kappa}} \left(\boldsymbol{z}_t - \boldsymbol{F}\boldsymbol{z}_{t-1}\right), \end{aligned}$$

where $\boldsymbol{e}_t = \left(\boldsymbol{y}_t - \boldsymbol{X}_t'\boldsymbol{\beta} - \boldsymbol{\Psi}_t^{\boldsymbol{w}}\boldsymbol{z}_t\right)$, and $\boldsymbol{\Sigma}_t = (\sigma_1 \boldsymbol{I}_{m_1,t}) \oplus \dots \oplus (\sigma_p \boldsymbol{I}_{m_p,t})$ is the covariance matrix of $\boldsymbol{\epsilon}_t$, and $\boldsymbol{Q}_{\boldsymbol{\kappa}} = \boldsymbol{Q}_{\kappa_1} \oplus \dots \oplus \boldsymbol{Q}_{\kappa_q}$ is the precision matrix of the innovation term $\boldsymbol{z}_t - \boldsymbol{F}\boldsymbol{z}_{t-1}$, the matrix $\boldsymbol{F} = (f_1 \boldsymbol{I}_{R_1}) \oplus \dots \oplus (f_q \boldsymbol{I}_{R_q})$ corresponds to the transition matrix and $\boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}_0$ are the vector mean and covariance matrix of $\boldsymbol{z}_0$, respectively. Thus, the conditional expectation of the complete-data log-likelihood given the current estimate of Π at pass (k) of the EM algorithm takes the form

$$Q(\Pi | \Pi^{(k)}) = \mathbb{E}_{\Pi^{(k)}} \left[-2\ell(\Pi; \boldsymbol{y}_{1:T}, \boldsymbol{x}_{1:T}, \boldsymbol{z}_{1:T}) | \boldsymbol{y}_{1:T}, \boldsymbol{X}_{1:T}\right], \tag{27}$$

where $\mathbb{E}_{\Pi^{(k)}}$ stands for the expected value considering the posterior distribution of the latent variables $\boldsymbol{z}_{1:T}$ given the observed data $(\boldsymbol{y}_{1:T}, \boldsymbol{X}_{1:T})$ and the current parameter estimate $\Pi^{(k)}$.

Applying the conditional expectation, it follows that

$$\begin{aligned} Q(\Pi | \Pi^{(k)}) &= \log|\boldsymbol{\Sigma}_0| + \mathrm{Tr}\left[\boldsymbol{\Sigma}_0^{-1}\left((\boldsymbol{z}_0^T - \boldsymbol{\mu}_0)(\boldsymbol{z}_0^T - \boldsymbol{\mu}_0)' + \boldsymbol{P}_0^T\right)\right] \\ &+ \sum_{t=1}^{T} \log|\boldsymbol{\Sigma}_t| + \sum_{t=1}^{T} \mathrm{Tr}\left[\boldsymbol{\Sigma}_t^{-1}\left(\left(\boldsymbol{y}_t - \boldsymbol{X}_t'\boldsymbol{\beta} - \boldsymbol{\Psi}_t^{\boldsymbol{w}}\boldsymbol{z}_t^T\right)\left(\boldsymbol{y}_t - \boldsymbol{X}_t'\boldsymbol{\beta} - \boldsymbol{\Psi}_t^{\boldsymbol{w}}\boldsymbol{z}_t^T\right)' + \boldsymbol{\Psi}_t^{\boldsymbol{w}}\boldsymbol{P}_t^T\boldsymbol{\Psi}^{\boldsymbol{w}}\right)\right] \\ &- T\log|\boldsymbol{Q}_{\boldsymbol{\kappa}}| + \mathrm{Tr}\left[\boldsymbol{Q}_k \sum_{t=1}^{T} (\boldsymbol{z}_t^T - \boldsymbol{F}\boldsymbol{z}_{t-1}^T)(\boldsymbol{z}_t^T - \boldsymbol{F}\boldsymbol{z}_{t-1}^T)'\right] \end{aligned}$$

$$+ \operatorname{Tr}\Big[\boldsymbol{Q}_{\boldsymbol{\kappa}} \sum_{t=1}^{T} \big(\boldsymbol{P}_t^T + \boldsymbol{F}\boldsymbol{P}_{t-1}^T\boldsymbol{F}' - \boldsymbol{F}\boldsymbol{P}_{t,t-1}^T - (\boldsymbol{P}_{t,t-1}^T)'\boldsymbol{F}'\big)\Big], \tag{28}$$

where the quantities $\boldsymbol{z}_t^t$ and $\boldsymbol{P}_t^t$ for $t = 0, \dots, T$ are obtained sequentially using the Kalman Filter equations, whose forward pass for $t =, 1, \dots, T$ consists of:

$$\begin{aligned}
\textbf{Prediction}: \quad \boldsymbol{z}_t^{t-1} &= \boldsymbol{F}^{(k)}\boldsymbol{z}_{t-1}^{t-1},\\
\boldsymbol{P}_t^{t-1} &= \boldsymbol{F}^{(k)}\boldsymbol{P}_{t-1}^{t-1}\boldsymbol{F}^{(k)\prime} + \boldsymbol{Q}_{\boldsymbol{\kappa}^{(k)}}^{-1},\\
\text{with} \quad \boldsymbol{z}_0^0 &= \boldsymbol{\mu}_0^{(k)}, \quad \boldsymbol{P}_0^0 = \boldsymbol{\Sigma}_0^{(k)}.\\
\textbf{Update(Filter)}: \quad \boldsymbol{\epsilon}_t &= \boldsymbol{y}_t - \boldsymbol{X}_t'\boldsymbol{\beta}^{(k)} - \boldsymbol{\Psi}_t^{\boldsymbol{w}^{(k)}}\boldsymbol{z}_t^{t-1},\\
\boldsymbol{\Sigma}_{\epsilon_t} &= \boldsymbol{\Psi}_t^{\boldsymbol{w}^{(k)}}\boldsymbol{P}_t^{t-1}(\boldsymbol{\Psi}_t^{\boldsymbol{w}})' + \boldsymbol{\Sigma}_t,\\
\boldsymbol{K}_t &= \boldsymbol{P}_t^{t-1}(\boldsymbol{\Psi}_t^{\boldsymbol{w}^{(k)}})'(\boldsymbol{\Sigma}_{\epsilon_t})^{-1},\\
\boldsymbol{z}_t^t &= \boldsymbol{z}_t^{t-1} + \boldsymbol{K}_t\boldsymbol{\epsilon}_t,\\
\boldsymbol{P}_t^t &= \boldsymbol{P}_t^{t-1} - \boldsymbol{K}_t\boldsymbol{\Psi}_t^{\boldsymbol{w}^{(k)}}\boldsymbol{P}_t^{t-1}.
\end{aligned}$$

and where the quantities $\boldsymbol{z}_t^T$ and $\boldsymbol{P}_t^T$ for $t = 0, \dots, T$ are obtained sequentially using the Kalman Smoothing equations, whose backwards pass for $t = T, \dots, 1$ consists of:

$$\begin{aligned}
\textbf{Smoothing}: \quad \boldsymbol{z}_T^T &= \boldsymbol{z}_T^T, \quad \boldsymbol{P}_T^T = \boldsymbol{P}_T^T, \quad \text{(for } t = T \text{ only)}\\
\boldsymbol{J}_{t-1} &= \boldsymbol{P}_{t-1}^{t-1}\boldsymbol{F}^{(k)\prime}(\boldsymbol{P}_t^{t-1})^{-1},\\
\boldsymbol{z}_{t-1}^T &= \boldsymbol{z}_{t-1}^{t-1} + \boldsymbol{J}_{t-1}(\boldsymbol{z}_t^T - \boldsymbol{z}_t^{t-1}),\\
\boldsymbol{P}_{t-1}^T &= \boldsymbol{P}_{t-1}^{t-1} + \boldsymbol{J}_{t-1}(\boldsymbol{P}_t^T - \boldsymbol{P}_t^{t-1})\boldsymbol{J}_t'.
\end{aligned}$$

The Smoothed lag-one covariance $\boldsymbol{P}_{t,t-1}^T$ for $t = 1, \dots, T$ is also obtain sequentially for $t = T, \dots, 2$ by

$$\begin{aligned}
\textbf{Smoothed lag-one covariance:} \quad \boldsymbol{P}_{T,T-1}^T &= (\boldsymbol{I} - \boldsymbol{K}_T\boldsymbol{\Psi}_{\boldsymbol{w}^{(k)}}^T)\boldsymbol{F}\boldsymbol{P}_{T-1}^{T-1}, \quad \text{(for } t = T \text{ only)}\\
\boldsymbol{P}_{t,t-1}^T &= \boldsymbol{P}_{t,t-1}\boldsymbol{J}_{t-2}' + \boldsymbol{J}_{t-2}(\boldsymbol{P}_{t-1,t-1}^T - \boldsymbol{F}\boldsymbol{P}_{t-1}^{t-1})\boldsymbol{J}_{t-2}'.
\end{aligned}$$

Now, we differentiate $Q(\Pi|\Pi^{(k)})$ from Eq. (28) concerning $\boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{w}, \boldsymbol{f}$ to derive closed-form expressions for the values that minimise the function. Meanwhile, the objective function, which must be minimised numerically, provides an estimate for κ_i for $i = 1, \dots, q$.

Denote by $\boldsymbol{e}_i$ the $q-$dimensional vector with all entries equal to zero except for the $i-$th that is equal to 1. Moreover, for $i = 1, \dots, p$, define $\boldsymbol{\Psi}_{i,t} = \boldsymbol{\Psi}_{i1,t} \oplus \cdots \oplus \boldsymbol{\Psi}_{iq,t}$ that is a $(qm_{i,t}) \times (qR)$ matrix.

We obtain:

$$\boldsymbol{\beta} = \left(\sum_{t=1}^{T} \boldsymbol{X}_t'(\boldsymbol{\Sigma}_t^{(k)})^{-1}\boldsymbol{X}_t\right)^{-1}\left(\sum_{t=1}^{T} \boldsymbol{X}_t'(\boldsymbol{\Sigma}_t^{(k)})^{-1}\left(\boldsymbol{y}_t - \boldsymbol{\Psi}_t^{\boldsymbol{w}^{(k)}}\boldsymbol{z}_t^T\right)\right). \tag{29}$$

For $i = 1, \dots, p$, update $(\sigma_i^2)^{(k+1)}$ as:

$$\begin{aligned}
(\sigma_i^2)^{(k+1)} = \left(\sum_{t=1}^{T} m_{i,t}\right)^{-1} \sum_{t=1}^{T} \operatorname{Tr}\Bigg\{ &\left(\boldsymbol{y}_{i,t} - \boldsymbol{X}_{i,t}'\boldsymbol{\beta}^{(k)} - \left[w_{i1}^{(k)}\boldsymbol{\Psi}_{i1}^t, \dots, w_{iq}^{(k)}\boldsymbol{\Psi}_{iq}^t\right]\boldsymbol{z}_t^T\right)\\
&\times \left(\boldsymbol{y}_{i,t} - \boldsymbol{X}_{i,t}'\boldsymbol{\beta}^{(k)} - \left[w_{i1}^{(k)}\boldsymbol{\Psi}_{i1}^t, \dots, w_{iq}^{(k)}\boldsymbol{\Psi}_{iq}^t\right]\boldsymbol{z}_t^T\right)'\\
&+ \left[w_{i1}^{(k)}\boldsymbol{\Psi}_{i1}^t, \dots, w_{iq}^{(k)}\boldsymbol{\Psi}_{iq}^t\right]\boldsymbol{P}_t^T\left[w_{i1}^{(k)}\boldsymbol{\Psi}_{i1}^t, \dots, w_{iq}^{(k)}\boldsymbol{\Psi}_{iq}^t\right]'\Bigg\}.
\end{aligned} \tag{30}$$

Continuing, we have

$$f_i = \frac{\operatorname{Tr}\left[\boldsymbol{Q}_{\kappa_i^{(k)}}\sum_{t=1}^{T}(\boldsymbol{e}_i' \otimes \boldsymbol{I}_R)(\boldsymbol{P}_{t,t-1}^T + \boldsymbol{z}_{t-1}^T\boldsymbol{z}_t^{T\prime})(\boldsymbol{e}_i \otimes \boldsymbol{I}_R)\right]}{\operatorname{Tr}\left[\boldsymbol{Q}_{\kappa_i^{(k)}}\sum_{t=1}^{T}(\boldsymbol{e}_i' \otimes \boldsymbol{I}_R)(\boldsymbol{P}_{t-1}^T + \boldsymbol{z}_{t-1}^T\boldsymbol{z}_{t-1}^{T\prime})(\boldsymbol{e}_i \otimes \boldsymbol{I}_R)\right]}. \tag{31}$$

Moreover,

$$\boldsymbol{\mu}_0 = \boldsymbol{z}_0^T, \quad \text{and} \quad \boldsymbol{\Sigma}_0 = \boldsymbol{P}_0^T.$$

Finally, to derive the closed-form expression for $\boldsymbol{w}$, proceed as follows.

Let us consider the quantity in the likelihood involving the only $\boldsymbol{w}_i = (w_{i1}, \dots, w_{iq})'$, that is:

$$\sum_{t=1}^{T}\mathbb{E}\left[\left\|\boldsymbol{y}(\mathcal{S}_{i,t}, t) - \boldsymbol{X}^i(\mathcal{S}_{i,t}), t - \sum_{j=1}^{q} w_{ij}(\boldsymbol{e}_j' \otimes \boldsymbol{I}_{m_i,t})\boldsymbol{\Psi}_{i,t}\boldsymbol{z}_t\right\|^2 \Big| \boldsymbol{y}_{1:T}, \boldsymbol{X}_{1:T}\right],$$

where $\|\cdot\|$ denotes the euclidean norm. Observe that

$$\sum_{j=1}^{q} w_{ij}(\boldsymbol{e}_j' \otimes \boldsymbol{I}_{m_i,t})\boldsymbol{\Psi}_{i,t}\boldsymbol{z}_t = \left[(\boldsymbol{e}_1' \otimes \boldsymbol{I}_{m_i,t})\boldsymbol{\Psi}_{i,t}\boldsymbol{z}_t, \dots, (\boldsymbol{e}_j' \otimes \boldsymbol{I}_{m_i,t})\boldsymbol{\Psi}_{i,t}\boldsymbol{z}_t\right]\boldsymbol{w}_i.$$

Therefore, we have to minimise in $\boldsymbol{w}_i$ the quantity

$$\begin{aligned}
&-2\sum_{t=1}^{T}\left(\boldsymbol{y}(S_{i,t},t) - \boldsymbol{X}^i S_{i,t},t)\right)' \mathbb{E}\left[(\boldsymbol{e}_1' \otimes \boldsymbol{I}_{m_i,t})\boldsymbol{\Psi}_{i,t}\boldsymbol{z}_t, \dots, (\boldsymbol{e}_j' \otimes \boldsymbol{I}_{m_i,t})\boldsymbol{\Psi}_{i,t}\boldsymbol{z}_t | \boldsymbol{y}_{1:T}, \boldsymbol{X}_{1:T}\right]\boldsymbol{w}_i \\
&+ \boldsymbol{w}_i'\mathbb{E}\Bigg[\left[(\boldsymbol{e}_1' \otimes \boldsymbol{I}_{m_i,t})\boldsymbol{\Psi}_{i,t}\boldsymbol{z}_t, \dots, (\boldsymbol{e}_j' \otimes \boldsymbol{I}_{m_i,t})\boldsymbol{\Psi}_{i,t}\boldsymbol{z}_t\right]' \\
&\qquad \times \left[(\boldsymbol{e}_1' \otimes \boldsymbol{I}_{m_i,t})\boldsymbol{\Psi}_{i,t}\boldsymbol{z}_t, \dots, (\boldsymbol{e}_j' \otimes \boldsymbol{I}_{m_i,t})\boldsymbol{\Psi}_{i,t}\boldsymbol{z}_t\right] \Bigg| \boldsymbol{y}_{1:T}, \boldsymbol{X}_{1:T}\Bigg]\boldsymbol{w}_i.
\end{aligned} \tag{32}$$

The first term in (32) is equivalent to $-2\boldsymbol{g}_i'\boldsymbol{w}_i$, where

$$\boldsymbol{g}_i' = \sum_{t=1}^{T}\left(\boldsymbol{y}(S_{i,t},t) - \boldsymbol{X}^i(S_{i,t},t)\boldsymbol{\beta}^{(k)}\right)' \left[(\boldsymbol{e}_1' \otimes \boldsymbol{I}_{m_i,t})\boldsymbol{\Psi}_{i,t}\boldsymbol{z}_t^T, \dots, (\boldsymbol{e}_q' \otimes \boldsymbol{I}_{m_i,t})\boldsymbol{\Psi}_{i,t}\boldsymbol{z}_t^T\right]. \tag{33}$$

The second term in (32) is equivalent to $\boldsymbol{w}_i'\boldsymbol{R}_i\boldsymbol{w}_i$, where $\boldsymbol{R}_i = [r_{i,jk}]_{j,k=1}^{q}$ is a $q \times q$ matrix with

$$\begin{aligned}
r_{i,jk} &= \sum_{t=1}^{T}\mathbb{E}\left[\boldsymbol{z}_t'\boldsymbol{\Psi}_{i,t}'(\boldsymbol{e}_j' \otimes \boldsymbol{I}_{m_i,t})'(\boldsymbol{e}_k' \otimes \boldsymbol{I}_{m_i,t})\boldsymbol{\Psi}_{i,t}\boldsymbol{z}_t | \boldsymbol{y}_{1:T}, \boldsymbol{X}_{1:T}\right] \\
&= \sum_{t=1}^{T}\boldsymbol{z}_t^{T'}\boldsymbol{\Psi}_{i,t}'(\boldsymbol{e}_j' \otimes \boldsymbol{I}_{m_i,t})'(\boldsymbol{e}_k' \otimes \boldsymbol{I}_{m_i,t})\boldsymbol{\Psi}_{i,t}\boldsymbol{z}_t^T + \operatorname{Tr}\left(\boldsymbol{\Psi}_{i,t}'(\boldsymbol{e}_j' \otimes \boldsymbol{I}_{m_i,t})'(\boldsymbol{e}_k' \otimes \boldsymbol{I}_{m_i,t})\boldsymbol{\Psi}_{i,t}\boldsymbol{P}_t^T\right) \\
&= \sum_{t=1}^{T}\operatorname{Tr}\left(\boldsymbol{\Psi}_{i,t}'\left((\boldsymbol{e}_j\boldsymbol{e}_k')\otimes \boldsymbol{I}_{m_i,t}\right)\boldsymbol{\Psi}_{i,t}\left(\boldsymbol{z}_t^T\boldsymbol{z}_t^{T'} + \boldsymbol{P}_t^T\right)\right).
\end{aligned}$$

Therefore, we have to minimise

$$-2\boldsymbol{g}_i'\boldsymbol{w}_i + \boldsymbol{w}_i'\boldsymbol{R}_i\boldsymbol{w}_i, \tag{34}$$

and therefore

$$\boldsymbol{w}_i^{(k+1)} = \boldsymbol{R}_i^{-1}\boldsymbol{g}_i. \tag{35}$$

For the updating the value of κ_i we numerically minimise the function $g_i^{(k)} : \mathbb{R}_+ \to \mathbb{R}_+$ defined as

$$\begin{aligned}
g_i(\kappa_i) = -T\log|\boldsymbol{Q}_{\boldsymbol{\kappa}_i}| &+ \operatorname{Tr}\Big[\boldsymbol{Q}_{\boldsymbol{\kappa}_i}\sum_{t=1}^{T}(\boldsymbol{e}_i' \otimes \boldsymbol{I}_R)(\boldsymbol{z}_t^T - \boldsymbol{F}^{(k)}\boldsymbol{z}_{t-1}^T)(\boldsymbol{z}_t^T - \boldsymbol{F}^{(k)}\boldsymbol{z}_{t-1}^T)'(\boldsymbol{e}_i \otimes \boldsymbol{I}_R)\Big] \\
&+ \operatorname{Tr}\Big[\boldsymbol{Q}_{\boldsymbol{\kappa}_i}\sum_{t=1}^{T}(\boldsymbol{e}_i' \otimes \boldsymbol{I}_R)\big(\boldsymbol{P}_t^T + \boldsymbol{F}^{(k)}\boldsymbol{P}_{t-1}^T\boldsymbol{F}^{(k)\prime} - \boldsymbol{F}^{(k)}\boldsymbol{P}_{t,t-1}^T - (\boldsymbol{P}_{t,t-1}^T)'\boldsymbol{F}^{(k)\prime}\big)(\boldsymbol{e}_i \otimes \boldsymbol{I}_R)\Big],
\end{aligned}$$

therefore

$$\kappa_i^{(k+1)} = \arg\min_{\kappa_i} g_i^{(k)}(\kappa_i).$$

Appendix F. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.spasta.2026.100971.

References

Andrej, J., Atallah, N., Bäcker, J.-P., Camier, J.-S., Copeland, D., Dobrev, V., Dudouit, Y., Duswald, T., Keith, B., Kim, D., et al., 2024. High-performance finite elements with MFEM. Int. J. High Perform. Comput. Appl. 38 (5), 447–467.

Bai, J., Wang, P., 2015. Identification and Bayesian estimation of dynamic factor models. J. Bus. Econom. Statist. 33 (2), 221–240.

Bevilacqua, M., Fassò, A., Gaetan, C., Porcu, E., Velandia, D., 2016. Covariance tapering for multivariate Gaussian random fields estimation. Stat. Methods Appl. 25 (1), 21–37.

Bevilacqua, M., Mateu, J., Porcu, E., Zhang, H., Zini, A., 2010. Weighted composite likelihood-based tests for space-time separability of covariance functions. Stat. Comput. 20, 283–293.

Bolin, D., Wallin, J., 2020. Multivariate type G Matérn stochastic partial differential equation random fields. J. R. Stat. Soc. Ser. B Stat. Methodol. 82 (1), 215–239.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M.J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., Zhang, Q., 2018. JAX: composable transformations of Python+NumPy programs.

Brenner, S.C., 2008. The Mathematical Theory of Finite Element Methods. Springer.

Calculli, C., Fassò, A., Finazzi, F., Pollice, A., Turnone, A., 2015. Maximum likelihood estimation of the multivariate hidden dynamic geostatistical model with application to air quality in Apulia, Italy. Environmetrics 26 (6), 406–417.
Cameletti, M., Lindgren, F., Simpson, D., Rue, H., 2013. Spatio-temporal modeling of particulate matter concentration through the SPDE approach. AStA Adv. Stat. Anal. 97, 109–131.
Ciarlet, P.G., 2002. The Finite Element Method for Elliptic Problems. SIAM.
Clarotto, L., Allard, D., Romary, T., Desassis, N., 2024. The SPDE approach for spatio-temporal datasets with advection and diffusion. Spat. Stat. 100847.
Cressie, N., 1985. Fitting variogram models by weighted least squares. J. Int. Assoc. Math. Geol. 17, 563–586.
Cressie, N., Johannesson, G., 2008. Fixed rank kriging for very large spatial data sets. J. R. Stat. Soc. Ser. B Stat. Methodol. 70 (1), 209–226.
Cressie, N., Shi, T., Kang, E.L., 2010. Fixed rank filtering for spatio-temporal data. J. Comput. Graph. Statist. 19 (3), 724–745.
Cressie, N., Wikle, C.K., 2002. Space-time Kalman filter. Encycl. Environmetrics 4, 2045–2049.
Cressie, N., Wikle, C.K., 2011. Statistics for Spatio-Temporal Data. John Wiley & Sons.
Datta, A., Banerjee, S., Finley, A.O., Gelfand, A.E., 2016. Hierarchical nearest-neighbor Gaussian process models for large geostatistical datasets. J. Amer. Statist. Assoc. 111 (514), 800–812.
Dutta, C., Ravishanker, N., Basu, S., 2024. Modeling multivariate positive-valued time series using R-INLA. Appl. Stoch. Models Bus. Ind. 40 (4), 830–849. http://dx.doi.org/10.1002/asmb.2834.
Ern, A., Guermond, J.-L., 2004. Theory and Practice of Finite Elements, vol. 159, Springer.
Fasso, A., Finazzi, F., 2011. Maximum likelihood estimation of the dynamic coregionalization model with heterotopic data. Environmetrics 22 (6), 735–748.
Fassò, A., Finazzi, F., Ndongo, F., 2016. European population exposure to airborne pollutants based on a multivariate spatio-temporal model. J. Agric. Biol. Environ. Stat. 21, 492–511.
Fassò, A., Rodeschini, J., Fusta Moro, A., Shaboviq, Q., Maranzano, P., Cameletti, M., Finazzi, F., Golini, N., Ignaccolo, R., Otto, P., 2023. Agrimonia: a dataset on livestock, meteorology and air quality in the Lombardy region, Italy. Sci. Data 10 (1), 143.
Field, D.A., 1988. Laplacian smoothing and Delaunay triangulations. Commun. Appl. Numer. Methods 4 (6), 709–712.
Finazzi, F., Scott, E.M., Fassò, A., 2013. A model-based framework for air quality indices and population risk evaluation, with an application to the analysis of Scottish air quality data. J. R. Stat. Soc. Ser. C. Appl. Stat. 62 (2), 287–308.
Furrer, R., Genton, M.G., Nychka, D., 2006. Covariance tapering for interpolation of large spatial datasets. J. Comput. Graph. Statist. 15 (3), 502–523.
Fusta Moro, A., Fassò, A., 2025. AQCLIM GRINS dataset - the Italian daily dataset on air quality. http://dx.doi.org/10.5281/zenodo.15699805, Zenodo. (Accessed 26 July 2025).
Gaedke-Merzhäuser, L., Maillou, V., Avellaneda, F.R., Schenk, O., Luisier, M., Moraga, P., Ziogas, A.N., Rue, H., 2025. Accelerated spatio-temporal Bayesian modeling for multivariate Gaussian processes. arXiv preprint arXiv:2507.06938.
Hamilton, J.D., 2020. Time Series Analysis. Princeton University Press.
Harris, C.R., Millman, K.J., Van Der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., et al., 2020. Array programming with NumPy. Nature 585 (7825), 357–362.
Heaton, M.J., Datta, A., Finley, A.O., Furrer, R., Guinness, J., Guhaniyogi, R., Gerber, F., Gramacy, R.B., Hammerling, D., Katzfuss, M., et al., 2019. A case study competition among methods for analyzing large spatial data. J. Agric. Biol. Environ. Stat. 24, 398–425.
Jona Lasinio, G., Mastrantonio, G., Pollice, A., 2013. Discussing the "big n problem". Stat. Methods Appl. 22, 97–112.
Katzfuss, M., Guinness, J., 2021. A general framework for vecchia approximations of Gaussian processes. Statist. Sci. 36 (1), 124–141.
Katzfuss, M., Guinness, J., Gong, W., Zilber, D., 2020. Vecchia approximations of Gaussian-process predictions. J. Agric. Biol. Environ. Stat. 25 (3), 383–414.
Kaufman, C.G., Schervish, M.J., Nychka, D.W., 2008. Covariance tapering for likelihood-based estimation in large spatial data sets. J. Amer. Statist. Assoc. 103 (484), 1545–1555.
Krainski, E., Gómez-Rubio, V., Bakka, H., Lenzi, A., Castro-Camilo, D., Simpson, D., Lindgren, F., Rue, H., 2018. Advanced Spatial Modeling with Stochastic Partial Differential Equations Using R and INLA. Chapman and Hall/CRC.
Lewbel, A., 2019. The identification zoo: Meanings of identification in econometrics. J. Econ. Lit. 57 (4), 835–903.
Lindgren, F., 2023. Fmesher: Triangle meshes and related geometry tools. R package version 0.1, 5.
Lindgren, F., Bakka, H., Bolin, D., Krainski, E., Rue, H., 2024. A diffusion-based spatio-temporal extension of Gaussian Matérn fields. SORT: Stat. Oper. Res. Trans.
Lindgren, F., Rue, H., Lindström, J., 2011. An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. J. R. Stat. Soc. Ser. B Stat. Methodol. 73 (4), 423–498.
Liu, H., Ong, Y.-S., Shen, X., Cai, J., 2020. When Gaussian process meets big data: A review of scalable GPs. IEEE Trans. Neural Netw. Learn. Syst. 31 (11), 4405–4423.
Nychka, D., Bandyopadhyay, S., Hammerling, D., Lindgren, F., Sain, S., 2015. A multiresolution Gaussian process model for the analysis of large spatial datasets. J. Comput. Graph. Statist. 24 (2), 579–599.
Otto, P., Fassò, A., Maranzano, P., 2024. A review of regularised estimation methods and cross-validation in spatiotemporal statistics. Stat. Surv. 18, 299–340.
Quarteroni, A., Valli, A., 2008. Numerical Approximation of Partial Differential Equations, vol. 23, Springer Science & Business Media.
Rasmussen, C.E., 2003. Gaussian processes in machine learning. In: Summer School on Machine Learning. Springer, pp. 63–71.
Rodeschini, J., Fassò, A., Finazzi, F., Fusta Moro, A., 2024. Scenario analysis of livestock-related PM2. 5 pollution based on a new heteroskedastic spatiotemporal model. Socio-Econ. Plan. Sci. 96, 102053.
Rue, H., Held, L., 2005. Gaussian Markov Random Fields: Theory and Applications. Chapman and Hall/CRC.
Rue, H., Martino, S., Chopin, N., 2009. Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. J. R. Stat. Soc. Ser. B Stat. Methodol. 71 (2), 319–392.
Ruiz-Cárdenas, R., Krainski, E.T., Rue, H., 2012. Direct fitting of dynamic models using integrated nested Laplace approximations—INLA. Comput. Statist. Data Anal. 56 (6), 1808–1828.
Schmidt, J., Hennig, P., Nick, J., Tronarp, F., 2023. The rank-reduced Kalman filter: Approximate dynamical-low-rank filtering in high dimensions. Adv. Neural Inf. Process. Syst. 36, 61364–61376.
Shumway, R.H., Stoffer, D.S., Stoffer, D.S., 2000. Time Series Analysis and Its Applications, vol. 3, Springer.
Song, Y., Dai, W., Genton, M.G., 2024. Large-scale low-rank Gaussian process prediction with support points. J. Amer. Statist. Assoc. 1–12.
Stein, M.L., 1999. Interpolation of Spatial Data: Some Theory for Kriging. Springer Science & Business Media.
Vecchia, A.V., 1988. Estimation and model identification for continuous spatial processes. J. R. Stat. Soc. Ser. B Stat. Methodol. 50 (2), 297–312.
Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., et al., 2020. SciPy 1.0: fundamental algorithms for scientific computing in python. Nature Methods 17 (3), 261–272.
Wackernagel, H., 2003. Multivariate Geostatistics: An Introduction with Applications. Springer.
Whittle, P., 1954. On stationary processes in the plane. Biometrika 434–449.
Whittle, P., 1963. Stochastic-processes in several dimensions. Bull. Int. Stat. Inst. 40 (2), 974–994.
Xu, K., Wikle, C.K., 2007. Estimation of parameterized spatio-temporal dynamic models. J. Statist. Plann. Inference 137 (2), 567–588.
Zammit-Mangion, A., Cressie, N., 2021. FRK: An R package for spatial and spatio-temporal prediction with large datasets. J. Stat. Softw. 98, 1–48.
Zhang, H., 2004. Inconsistent estimation and asymptotically equal interpolations in model-based geostatistics. J. Amer. Statist. Assoc. 99 (465), 250–261.
Zhang, H., 2007. Maximum-likelihood estimation for multivariate spatial linear coregionalization models. Environmetrics: Off. J. Int. Environmetrics Soc. 18 (2), 125–139.