Posterior Inference for Sparse Hierarchical Non-stationary Models

Karla Monterrubio-Gómez[a], Lassi Roininen[b,1], Sara Wade[c,1,*], Theodoros Damoulas[a,e], Mark Girolami[d,e]

[a] *University of Warwick, UK*
[b] *LUT University, Finland*
[c] *University of Edinburgh, UK*
[d] *University of Cambridge, UK*
[e] *Alan Turing Institute, UK*

Abstract

Gaussian processes are valuable tools for non-parametric modelling, where typically an assumption of stationarity is employed. While removing this assumption can improve prediction, fitting such models is challenging. In this work, hierarchical models are constructed based on Gaussian Markov random fields with stochastic spatially varying parameters. Importantly, this allows for non-stationarity while also addressing the computational burden through a sparse banded representation of the precision matrix. In this setting, efficient Markov chain Monte Carlo (MCMC) sampling is challenging due to the strong coupling a posteriori of the parameters and hyperparameters. We develop and compare three adaptive MCMC schemes and make use of banded matrix operations for faster inference. Furthermore, a novel extension to higher dimensional input spaces is proposed through an additive structure that retains the flexibility and scalability of the model, while also inheriting interpretability from the additive approach. A thorough assessment of the efficiency and accuracy of the methods in nonstationary settings is presented for both simulated experiments and a computer emulation problem.

Keywords: Gaussian process, Multilevel models, Gaussian Markov random fields, MCMC, SPDE,
2000 MSC: 65C40, 62M05

*Corresponding author
Email addresses: k.monterrubio-gomez@warwick.ac.uk (Karla Monterrubio-Gómez), lassi.roininen@lut.fi (Lassi Roininen), sara.wade@ed.ac.uk (Sara Wade), t.damoulas@warwick.ac.uk (Theodoros Damoulas), m.girolami@cambridge.ac.uk (Mark Girolami)

[1] These authors contributed equally to this work.

1. Introduction

Gaussian processes are frequently utilised in constructing powerful nonparametric models, which are appealing due to their analytical properties. The flexibility and nonparametric nature of these models make them appropriate and useful in a wide range of applications. Gaussian process (GP) priors have been used in geostatistics [33] under the name of Kriging. They are also common in other applications; for instance, in atmospheric sciences [2], biology [49] and inverse problems [27]. A recent review and comparison of some available methods employing GPs is provided by Heaton et al. [24].

A large amount of research on GPs and their applications has focused on models where an assumption of stationarity for the process of interest is made. Nevertheless, this assumption is rarely realistic in practice and as a consequence, several approaches to introduce non-stationarity have been proposed [e.g. 1, 22, 29, 35, 47]. Although comparative evaluations show that removing the stationary assumption improves predictive accuracy [15, 22, 38], fitting such non-stationary models has proven to be challenging. This, combined with the well-known computational constraints of GP models, arising from storing covariance matrices, solving linear systems and computing determinants, poses important questions on how to efficiently perform Bayesian inference in non-stationary problems.

The stochastic partial differential equation (SPDE) approach introduced by Lindgren et al. [32] employs Gaussian Markov random fields (GMRFs) to ameliorate the computational burden of working with GPs and incorporates a non-stationary framework through spatially varying parameters that are modelled as a linear combination of basis functions. Similarly, Paciorek and Schervish [39] proposed a family of closed-form non-stationary covariance functions with spatially varying parameters modelled by a second latent GP prior. While recognised as a flexible construction, doing inference in a fully Bayesian framework becomes impractical due to the computational demands of such models. Moreover, standard Markov Chain Monte Carlo (MCMC) procedures require careful parameter tuning, exhibit mixing difficulties and require long runs to reach convergence [38, 39].

This work provides extensions of the SPDE formulation of non-stationary GPs initially introduced by Roininen et al. [45]. Such model is analogous to SPDE-based constructions in spatial interpolation [17, 18, 56], and to the non-stationary framework proposed by Paciorek and Schervish [39], where the spatially varying parameters are modelled as random objects. Specifically, our work incorporates and accounts for uncertainty in the measurement noise variance and hyperprior parameters and consider two hyperpriors for the spatially varying length-scale to account for different smoothness assumptions. Because, the hierarchical structure of these models, that we refer to as 2-level GPs, introduces strong dependencies and hence efficient sampling from the posterior distribution is problematic, we introduce and offer a comparative evaluation of three MCMC sampling schemes. The first corresponds to an adaptive Metropolis-within-Gibbs scheme. The second employs elliptical slice sampling

(ELL-SS) combined with re-parametrisations for decoupling the prior, hyper-prior, and hyperparameters. The third is a marginal sampler with ELL-SS for a re-parametrised length-scale process. More precisely, at the first level, the SPDE formulation provides a sparse factorization of the precision matrix of the non-stationary field. At the second level, we compare the exponential covariance function, which leads to a sparse precision matrix for the latent parameters of the non-stationary covariance matrix, with the squared exponential covariance function, which while popular, leads to a dense representation.

In addition, extensions of the 2-level GPs to higher dimensional input spaces are important and necessary in many applications. Existing approaches for two-dimensional settings are based on heavily parametrised models using spectral decompositions [38, 39, 43], basis function representations [28], or an isotropic assumption [26, 45]. Instead, we propose a novel extension based on additive GPs [12] that decomposes the function of interest in terms of low-dimensional functions, which are modelled as separable non-stationary processes. Important advantages include increased intrepretability and robustness to curse of dimensionality, while inheriting the appealing flexibility of 2-level GPs. The additive structure permits scalability, by taking advantage of the sparse banded precision matrices, low-dimensional representation, and efficient Kronecker algebra for the separable interaction terms. Moreover, it can capture long-range structures in the data. The choice of interaction terms may be application driven, and hyper-priors can be employed to determine their importance. In this case, the MCMC schemes can be extended through a Gibbs sampling framework. This extension provide an efficient method for data-dense problems in low dimensions but also enables using the construction for multidimensional (nD) problems with relatively sparse data, similar to [53].

The 2-level models studied here naturally extend to multiple levels to construct the deep GP models of Dunlop et al. [9]. Deep GPs have received increased interest in literature and proposals differ in how the layers are combined [e.g. 3, 7, 9, 25]. However, the key challenges, preventing wide-spread use of Deep GPs, include developing interpretable constructions that lack degeneracy [11] and efficient and scalable inference, despite the highly coupled layers and computational expense of GPs. The hierarchical construction considered here provides an interpretable structure for nonstationary problems, as well as a sparse framework to address the computational burden, providing a promising route to deeper constructions. Moreover, the developed methodology results in a non-stationary hierarchical construction that retains the flexibility of the model introduced by Paciorek and Schervish [39] but is computationally more efficient.

The paper is organised as follows. We start by summarising related work in Section 2. In Section 3, we present the sparse non-stationary hierarchical model for one-dimensional problems and describe the proposed sampling schemes in Section 4. Section 5 extends the model to higher dimensional input spaces, while retaining the computational benefits and flexibility. The experiments in Section 6 provide a complete empirical evaluation, with a study of the discretisation and sample size effects and performance for different signal types, as

well as a comparison with alternative GP models. Finally, Section 6.4 applies the methodology to a computer emulation problem for a NASA rocket booster vehicle.

2. Related work and background

We begin with a review of Gaussian process models, providing a connection between the non-stationary GPs of Paciorek and Schervish [39] and the SPDE formulation in Lindgren et al. [32] and Roininen et al. [45].

2.1. Gaussian process models

Let us denote by $\mathbf{y} \in \mathbb{R}^m$ noisy realisations of an unknown random process $\{z(\mathbf{x}), \mathbf{x} \in \mathbb{R}^d\}$. A standard GP regression model assumes

$$y_i = z(\mathbf{x}_i) + \varepsilon_i, \tag{2.1}$$

where ε_i is zero-mean Gaussian noise with variance σ_ε^2 and $z(\cdot)$ a Gaussian process. More precisely, the model can be written in a hierarchical form,

$$\begin{aligned} y_i &\sim \mathcal{N}(z(\mathbf{x}_i), \sigma_\varepsilon^2), \quad i = 1, \ldots, m, \\ z(\cdot) &\sim \mathcal{GP}\left(0, C_{\boldsymbol{\phi}}(\cdot, \cdot)\right), \\ (\boldsymbol{\phi}, \sigma_\varepsilon^2) &\sim \pi(\boldsymbol{\phi})\pi(\sigma_\varepsilon^2), \end{aligned} \tag{2.2}$$

where $C_{\boldsymbol{\phi}}(\cdot, \cdot)$ is a covariance function parametrised by $\boldsymbol{\phi}$ and must define a valid covariance matrix (symmetric and positive semi-definite). The covariance function encodes important properties of the process, such as its variation and smoothness. Stationary covariance functions only depend on the inputs $(\mathbf{x}_i, \mathbf{x}_j)$ through $(\mathbf{x}_i - \mathbf{x}_j)$ and are most often the default choice. Typical covariance functions include the stationary squared exponential (SE),

$$C^{\text{S}}(\mathbf{x}_i, \mathbf{x}_j) = \tau^2 \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\lambda^2}\right), \tag{2.3}$$

and the stationary Matérn family, formulated as

$$C^{\text{S}}(\mathbf{x}_i, \mathbf{x}_j) = \tau^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{\lambda}\right)^\nu K_\nu \left(\frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{\lambda}\right), \tag{2.4}$$

where $\Gamma(\cdot)$ is the gamma-function, $\nu > 0$ is the smoothness parameter, $\lambda > 0$ is the length-scale, $\tau^2 > 0$ is the magnitude or variance parameter, and K_ν denotes the modified Bessel function of the second kind of order ν.

However, the translation-invariance assumption of stationary covariance functions may be inappropriate for certain applications where the process is spatially dependent, such as, for problems in environmental, geospatial and urban sciences. In these cases, a non-stationary formulation of the model is desirable.

Paciorek and Schervish [39] introduced a family of non-stationary covariance functions,

$$C^{\text{NS}}(\mathbf{x}_i, \mathbf{x}_j) = \frac{\tau^2 |\Sigma(\mathbf{x}_i)|^{\frac{1}{4}} |\Sigma(\mathbf{x}_j)|^{\frac{1}{4}}}{|(\Sigma(\mathbf{x}_i) + \Sigma(\mathbf{x}_j))/2|^{\frac{1}{2}}} R\left(\sqrt{Q_{ij}}\right),$$

where $R(\cdot)$ is a stationary correlation function on $\mathbb{R}$; $\Sigma(\cdot)$ is a $d \times d$ spatially varying covariance matrix, referred to as a kernel matrix, which describes local anisotropies; and $Q_{ij} = (\mathbf{x}_i - \mathbf{x}_j)^{\text{T}} \left((\Sigma(\mathbf{x}_i) + \Sigma(\mathbf{x}_j))/2\right)^{-1} (\mathbf{x}_i - \mathbf{x}_j)$.

The non-stationary version of the Matérn covariance function is therefore,

$$C^{\text{NS}}(\mathbf{x}_i, \mathbf{x}_j) = \frac{\tau^2 2^{1-\nu} |\Sigma(\mathbf{x}_i)|^{\frac{1}{4}} |\Sigma(\mathbf{x}_j)|^{\frac{1}{4}}}{\Gamma(\nu) |(\Sigma(\mathbf{x}_i) + \Sigma(\mathbf{x}_j))/2|^{\frac{1}{2}}} \left(\sqrt{Q_{ij}}\right)^{\nu} K_\nu \left(\sqrt{Q_{ij}}\right), \tag{2.5}$$

with hyperparameters $\boldsymbol{\phi} = \{\Sigma(\cdot), \nu, \tau^2\}$. When employing this type of non-stationary covariance function in equation (2.2), we are required to infer the kernel matrices at every location where the process was observed. Paciorek and Schervish [39] modelled the kernel matrices as a continuous-parameter random process by utilising its spectral decomposition. Nonetheless, this approach results in computationally expensive inference [39, Section 5.1] even for one-dimensional problems. As a consequence, alternative approaches to model the spatially varying parameters have been proposed [31, 38, 42].

The closed-form kernel in equation (2.5) have given rise to different schemes in the literature to model non-stationary datasets. Firstly, Stein [50] extended the results from Paciorek and Schervish [39] and the work of Pintore and Holmes [40] to obtain an extremely flexible kernel, which corresponds to a generalisation of the non-stationary Matérn where all parameters are allowed to vary in space; however, he also pointed out that, even for a fixed ν, spatially varying τ^2 and $\Sigma(\cdot)$ leads to problems of consistent estimation in the parameters. Later, Kleiber and Nychka [30] developed further the work of Stein [50] by extending the kernel to multivariate settings, and more recently, Risser [42] derived a class of non-stationary kernels that enable the use of covariate information to drive non-stationarity. Here, we focus on the non-stationary family of kernels derived by Paciorek and Schervish [39], where the non-stationarity is introduced by allowing only one of the parameters, namely $\Sigma(\cdot)$, to vary in space. We note that for one-dimensional problems, the kernel matrices, $\Sigma(\cdot)$ in equation (2.5), are reduced to scalars, which we denote as $\ell(\cdot)$. In this setting, when modelling the spatially varying length-scale with a GP, the hierarchical formulation of the model is

$$\begin{aligned} y_i &\sim \mathcal{N}(z(x_i), \sigma_\varepsilon^2), \quad i = 1, \ldots, m, \\ z(\cdot) &\sim \mathcal{GP}\left(0, C^{\text{NS}}_{\boldsymbol{\phi}}(\cdot, \cdot)\right), \\ \log \ell(\cdot) &\sim \mathcal{GP}\left(\mu_\ell, C^{\text{S}}_{\boldsymbol{\varphi}}(\cdot, \cdot)\right), \\ (\tau^2, \boldsymbol{\varphi}, \sigma_\varepsilon^2, \mu_\ell) &\sim \pi(\tau^2)\pi(\boldsymbol{\varphi})\pi(\sigma_\varepsilon^2)\pi(\mu_\ell), \end{aligned} \tag{2.6}$$

where $C^{\text{NS}}_{\boldsymbol{\phi}}(\cdot, \cdot)$ is as in equation (2.5) and $C^{\text{S}}_{\boldsymbol{\varphi}}(\cdot, \cdot)$ is a stationary covariance function with parameters $\boldsymbol{\varphi}$. We note that the prior for the spatially varying

length-scale is assigned over a transformed parameter, defined as $u(\cdot) := \log \ell(\cdot)$, with μ_ℓ representing the a priori constant mean of the log length-scale process.

Efficient sampling from the posterior is challenging and the computational burden introduced by the spatially varying parameter is noticeable even in one-dimensional problems [26, 39]. These difficulties arise from different sources. First, the computational complexity inherited from dense covariance matrices makes the model unsuitable for large datasets. Second, the latent processes and hyperparameters tend to be strongly coupled, leaving vanilla MCMC schemes inefficient. Finally, as in a stationary formulation, the model is sensitive to the choice of hyperparameters, $\boldsymbol{\varphi}$, and therefore these must be inferred [38].

2.2. SPDE formulation of Matérn fields

Lindgren et al. [32] showed that Gaussian Markov random fields can be presented equivalently as stochastic partial differential equations. By fixing $\nu = 2 - d/2$, a GP with stationary Matérn covariance (2.4) and a Markov property can be defined through

$$\left(1 - \lambda^2 \Delta\right) z = \tau \sqrt{\lambda^d} w, \tag{2.7}$$

where $\Delta := \sum_{k=1}^{d} \partial^2 / \partial x_k^2$ is the Laplace operator, w is white noise on $\mathbb{R}^d$, and $\mathrm{Var}(w) = \Gamma(\nu + d/2)(4\pi)^{d/2}/\Gamma(\nu)$.

Analogous to the construction of Paciorek and Schervish [39] for non-stationary covariance functions with spatially varying length-scales, Roininen et al. [45] derive an SPDE formulation for non-stationary Matérn fields,

$$\left(1 - \ell(\cdot)^2 \Delta\right) z = \tau \sqrt{\ell(\cdot)^d} w, \tag{2.8}$$

where $\ell(\cdot)$ is a spatially varying length-scale, that is modelled as a log-transformed continuous-parameter GP in the hyperprior in equation (2.6). An alternative formulation was proposed by Lindgren et al. [32, Section 3.2], where spatially varying parameters were modelled through a basis function representation. Such a choice gives computational advantages, through a lower dimensional parameter space. However, this requires selecting the number of basis functions, and the ability to flexibly recover changes in the length-scale strongly depends on this choice.

A finite-dimensional approximation of our continuous-parameter model (2.8) can be written in vector-matrix format as $L(\boldsymbol{\ell})\mathbf{z} = \mathbf{w}$, where $L(\boldsymbol{\ell})$ is a sparse matrix depending on $\ell_j := \ell(jh)$, with h denoting the discretisation step in a chosen finite difference approximation. This model is constructed in such a way that the finite-dimensional approximation converges to the continuous-parameter model (2.8) in the discretisation limit $h \to 0$ (for proofs, see Roininen et al. [45]). This property guarantees that irrespective of the choice of h, the posteriors, and hence also the estimators, on different meshes, that are dense enough, are essentially the same.

We note that employing a GP to model $\ell(\cdot)$ results in a similar construction to that discussed in Section 2.1 and can be rephrased through

$$\mathbf{y} = \mathcal{A}z + \boldsymbol{\varepsilon} \approx A\mathbf{z} + \boldsymbol{\varepsilon}, \tag{2.9}$$

where $\mathcal{A}$ represents a linear mapping from some function space to a finite-dimensional space $\mathbb{R}^m$ and $\boldsymbol{\varepsilon} \in \mathbb{R}^m$ is assumed to be zero-mean Gaussian noise with variance $\sigma_\varepsilon^2 I_m$, which is independent of z. For computational reasons, we discretise this equation, such that $\mathcal{A}z \approx A\mathbf{z}$, obtaining the right hand side of equation (2.9), where $A \in \mathbb{R}^{m\times n}$ is a known matrix and $\mathbf{z} \in \mathbb{R}^n$ with $\mathbf{z} \sim \mathcal{N}(0, C_{\boldsymbol{\phi}}^{\text{NS}})$. In this case, through the matrix A, we are able to define the grid resolution of the latent fields. In particular, for more rough processes, we may be interested in finer resolutions, while for smooth functions, a sparse grid may be sufficient to obtain an accurate representation.

In the next sections, we extend the work of Roininen et al. [45], which was limited, with respect to the statistical model, to a simpler model that did not consider inference of the measurement noise variance and the length-scale hyper-parameter. Also, their sampling was based on Metropolis-within-Gibbs, which, as we will show, is unsuitable for deep hierarchical models and bigger datasets. In this paper, specifically, we explore different hyperprior models, discusse and compare MCMC algorithms to do inference with these types of models, and present a novel efficient way to extend the model to higher dimensions.

3. Sparse non-stationary hierarchical models

Our aim is to decompose the inverse covariance matrix $(C_{\mathbf{u}}^{\text{NS}})^{-1} := Q_{\mathbf{u}} = L(\mathbf{u})^{\text{T}} L(\mathbf{u})$, where $L(\mathbf{u})$ is a sparse matrix that depends on the log length-scale parameters $\mathbf{u} = \log(\boldsymbol{\ell})$. The required decomposition can be achieved employing the SPDE approach from Section 2.2. An explicit hierarchical formulation of the model is

$$\begin{aligned} \mathbf{y} \mid \mathbf{z}, \sigma_\varepsilon^2 &\sim \mathcal{N}(A\mathbf{z}, \sigma_\varepsilon^2 I_m), \\ \mathbf{z} \mid \mathbf{u} &\sim \mathcal{N}\left(0, Q_{\mathbf{u}}^{-1}\right), \\ \mathbf{u} \mid \lambda &\sim \mathcal{N}\left(\boldsymbol{\mu}_\ell, C_\lambda\right), \\ (\sigma_\varepsilon^2, \lambda) &\sim \pi(\sigma_\varepsilon^2)\pi(\lambda), \end{aligned} \tag{3.1}$$

where $\boldsymbol{\mu}_\ell$ denotes the n-dimensional vector with all elements equal to μ_ℓ. As both the length-scale and magnitude parameters cannot be estimated consistently [57], we use the observe data to set the magnitude and mean of both the stationary and non-stationary processes to improve identifiability, with full details provided in the Supplementary Material. The key component of the model is $Q_{\mathbf{u}}$, the inverse covariance of the GMRF employed to represent the non-stationary GP. This precision matrix depends on $\mathbf{u}$, which is assumed to be a constant-mean GP that describes the spatially varying log length-scale, and λ denotes the length-scale parameter of the covariance function that describes

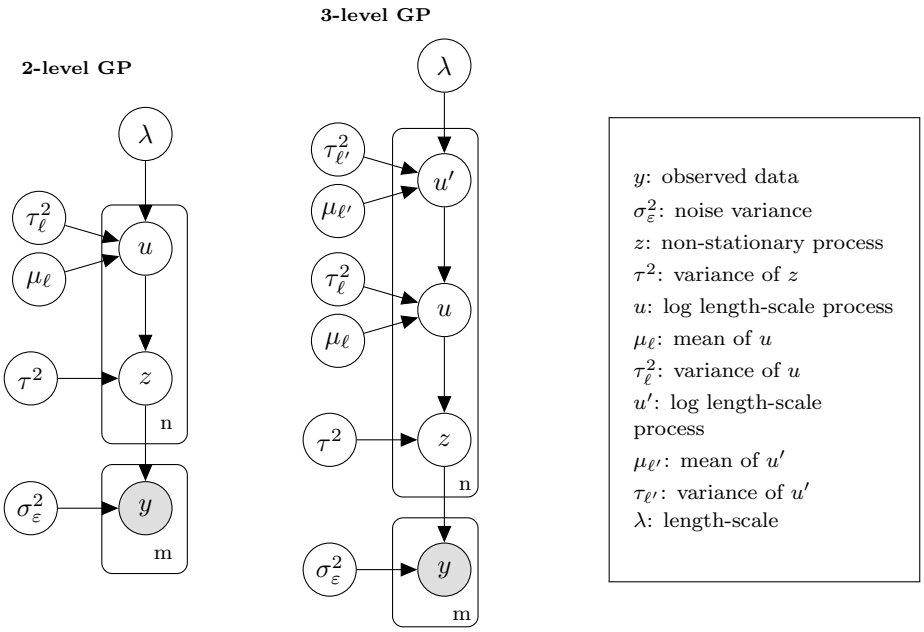


Figure 1: Plate diagram for a non-stationary hierarchical model.

the properties of the log length-scale process. A plate diagram of this model is given in Figure 3 (left).

We highlight that the SPDE formulation employed (2.7) considers periodic boundary conditions, which can lead to undesirable effects in the edges of the estimators. In order to correct a possible boundary effect, one can add points around the boundary. This domain extension offers also a possible benefit in the sparse structure of $L(\boldsymbol{\ell})$. By construction, the matrix $L(\boldsymbol{\ell})$ is a cyclic tridiagonal matrix, and while Sherman-Morrison formula can be applied to solve this type of systems efficiently (e.g. Seiler and Seiler [48]), we can simply neglect the matrix elements in the corners once we have applied domain extension and take advantage of the resulting tridiagonal structure.

In the following, we are interested on exploring the properties and behaviour of the model and algorithms under two extreme smoothness assumptions of the length-scale process. To do this, we analyse the methods under two different types of hyperpriors for $\mathbf{u}$. On the one hand, we introduce strong prior smoothness assumptions by using a squared exponential covariance. On the other hand, we explore rough hyperpriors, thorough an autoregressive AR(1) model.

The latter representation adds further computational gains to the model. In addition, notice that we are free to assign an inhomogeneous Matérn field for the log length-scale process, introducing more flexibility to the model. A graphical representation of this type of 3-level construction is given to the right of Figure 3. For simplicity, we focus on the 2-level case, when the parameters of the log length-scale process are restricted to be constant along the input space.

AR(1) hyperprior. A hyperprior with sample paths smoother than white noise is needed, otherwise different discretisations of z may affect the posterior estimates [45]. One such process is the Ornstein-Uhlenbeck, a member of the stationary Matérn family (equation (2.4)), with exponential covariance function obtained by setting $\nu = 1/2$. The Ornstein-Uhlenbeck has non-differentiable sample paths, allowing quick changes in the behaviour of the log length-scale process. It is the continuous-time counterpart of the first-order autoregressive model AR(1) given by $u_j = \beta u_{j-1} + e_j$ and $e_j \sim \mathcal{N}(0, \sigma^2)$, where u_j is on an uniform lattice $t_j := jh$, $j \in \mathbb{Z}$ with discretisation step h. Without a proof, we note that the AR(1) has an exponential autocovariance for all $\beta > 0$ except for $\beta = 1$ which corresponds to Gaussian random walk, i.e. Brownian motion. While the stable AR(1) requires that $\beta < 1$, this is not a necessary condition here, as our goal is in forming covariance matrices. Let us denote by $a_0 := 1/\sigma$ and $a_1 := \beta/\sigma$. Then, we can construct the inverse of the exponential covariance matrix $(C_\lambda^{\mathrm{s}})^{-1} := Q_\lambda = L(\lambda)^{\mathrm{T}} L(\lambda)$, where $L(\lambda)$ is a sparse matrix that depends on λ and τ_ℓ. More precisely, $L(\lambda)$ is a banded matrix, with nonzero elements only on the main diagonal given by $(a_0, \ldots, a_0, 1)$ and the first diagonal above this given by $(a_1, \ldots, a_1)$. The coefficients are defined as

$$a_0 = (\sqrt{h/\lambda} + \sqrt{h/\lambda + 4\lambda/h})/\tau_\ell\sqrt{8} \text{ and } a_1 = (\sqrt{h/\lambda} - \sqrt{h/\lambda + 4\lambda/h})/\tau_\ell\sqrt{8}.$$

Hence, we have a sparse representation for the hyperprior precision matrix, and the banded structure in $L(\lambda)$ offers important computational advantages when evaluating $\mathcal{N}(\mathbf{u} \mid \boldsymbol{\mu}_\ell, Q_\lambda^{-1})$, as the required determinant computations, matrix multiplications, and system of equations can be significantly simplified. We emphasise that higher-order autoregressive priors can be incorporated to the model by modify the smoothness prior assumption of the length-scale process. For a general AR(k) prior the bandwidth of $L(\lambda)$ will be $k + 1$ and therefore, the computational complexity will increase accordingly.

SE hyperprior. In contrast to the AR(1) hyperprior, we have the squared exponential hyperprior (equation (2.3)) for C_λ. This covariance function, also referred to as the radial basis function (RBF), is recovered when $\nu \to \infty$ in the stationary Matérn covariance of equation (2.4). Sample paths from a SE are infinitely differentiable and consequently very smooth. Therefore, when employing a SE hyperprior for the length-scale process, we introduce strong prior smoothness assumptions on how the correlation of the non-stationary process changes with distance. We note that for the SE hyperprior, the precision matrix is dense and therefore, comes at an increased computational cost.

4. Inference for one-dimensional input spaces

In order to efficiently draw samples from the posterior distributions of interest, we explore three MCMC sampling approaches. The first draws samples from the multidimensional vector $\mathbf{u}$ through an adaptive Metropolis-within-Gibbs algorithm. The second employs ancillary augmentation [55] over $\mathbf{z}$ and $\mathbf{u}$ and

uses elliptical slice sampling [ELL-SS, 37] over the re-parametrised log length-scale process. The third integrates out the non-stationary process, resulting in a marginal sampler that draws from $\mathbf{u}$ by combining ancillary augmentation and ELL-SS to break the correlation between $\mathbf{u}$ and λ.

4.1. Metropolis-within-Gibbs (MWG)

This sampling scheme resembles the algorithm proposed in Roininen et al. [45] with extensions to infer other model parameters and for other length-scale hyperpriors. More precisely, we include adaptive random walks [44] for the noise variance, length-scale hyperparameter and log length-scale process in order to obtain an scheme that is free of parameter tuning. Moreover, we discuss in detail the computational complexity of the algorithm, which is useful for comparison with the other algorithms and methods. The procedure is detailed in Supplementary Algorithm S.1.

The MWG framework updates the log length-scale process at each location individually and, regardless of the hyperprior employed, offers computational gains due to the fact that when proposing a single element of the log length-scale process u_k^*, for $k = 1, \ldots, n$, the log-ratio of the prior density of $\mathbf{z}$ used in the acceptance probability simplifies to

$$\log\left(\frac{\mathcal{N}(\mathbf{z} \mid 0, Q_{\mathbf{u}^*}^{-1})}{\mathcal{N}(\mathbf{z} \mid 0, Q_{\mathbf{u}}^{-1})}\right) = \log\det(L(\mathbf{u}^*)L(\mathbf{u})^{-1}) - \frac{1}{2}\mathbf{z}^{\mathrm{T}}\left(L(\mathbf{u}^*)^{\mathrm{T}}L(\mathbf{u}^*) - L(\mathbf{u})^{\mathrm{T}}L(\mathbf{u})\right)\mathbf{z}.$$

Here $\mathbf{u}^*$ is the proposed log length-scale vector, obtained by updating the kth element of $\mathbf{u}$ to u_k^*, and combined with pentadiagonal form of the precision matrix, derived from multiplication of tridiagonal matrices $Q_{\mathbf{u}} = L(\mathbf{u})^{\mathrm{T}}L(\mathbf{u})$, results in a reduced computational complexity of the quadratic term in the log-ratio from $O(n^2)$ to $O(1)$. Moreover, the log-determinant can be computed through numerically stable and inexpensive operations; for details, see Roininen et al. [45, Section 6]. Similarly, the log-ratio of the prior density of $\mathbf{u}$ simplifies to

$$\log\left(\frac{\mathcal{N}(\mathbf{u}^* \mid \boldsymbol{\mu}_\ell, C_\lambda)}{\mathcal{N}(\mathbf{u} \mid \boldsymbol{\mu}_\ell, C_\lambda)}\right) = -\frac{1}{2}\left([(u_k^*)^2 - u_k^2]Q_{\lambda\,k,k} + \sum_{j \neq k}[u_k^* - u_k]u_j Q_{\lambda\,k,j}\right),$$

where $Q_{\lambda\,k,j}$ denotes the (k, j) element of the matrix Q_λ. Further computational gains are possible when we utilise the AR(1) hyperprior, as the tridiagonal form $Q_\lambda = L(\lambda)^{\mathrm{T}}L(\lambda)$, resulting from the sparse AR(1) construction of $L(\lambda)$, reduces this operation from $O(n)$ to $O(1)$.

Additionally, when proposing a new hyperparameter λ^*, we must evaluate

$$\log\left(\frac{\mathcal{N}(\mathbf{u} \mid \boldsymbol{\mu}_\ell, C_{\lambda^*})}{\mathcal{N}(\mathbf{u} \mid \boldsymbol{\mu}_\ell, C_\lambda)}\right) = \frac{1}{2}\log\det(Q_{\lambda^*}Q_\lambda^{-1}) - \frac{1}{2}(\mathbf{u} - \boldsymbol{\mu}_\ell)^{\mathrm{T}}(Q_\lambda - Q_{\lambda^*})(\mathbf{u} - \boldsymbol{\mu}_\ell).$$

For the SE hyperprior, this requires the inversion of a dense $n \times n$ matrix, while the tridiagonal form of Q_λ for the AR(1) hyperprior makes this considerably cheaper by reducing the computational complexity of this log-ratio term from $O(n^3)$ to $O(n)$. In addition, our simulation studies show that this algorithm does not perform well when the hyperprior for $u(\cdot)$ has strong smoothness assumptions, such as those induced by employing a SE covariance function. This flaw motives us to explore alternative algorithms.

4.2. Whitened elliptical slice sampling (w-ELL-SS)

Elliptical slice sampling is a state-of-the-art MCMC algorithm for latent Gaussian models [37]. Here, we combine this sampling algorithm with ancillary augmentation or *whitening* [55], which represents a computationally cheap and effective strategy to break the correlation between the prior and its corresponding hyperparameters [14, 36].

We can equivalently define the unknown function as $\mathbf{z} = L(\mathbf{u})^{-1}\boldsymbol{\xi}$ with $\boldsymbol{\xi} \sim \mathcal{N}(0, I_n)$ and the log length-scale vector as $\mathbf{u} = R_\lambda \boldsymbol{\zeta} + \boldsymbol{\mu}_\ell$ with $\boldsymbol{\zeta} \sim \mathcal{N}(0, I_n)$. For the AR(1) hyperprior, $R_\lambda := L(\lambda)^{-1}$; whereas, for the SE hyperprior, we define R_λ to be the lower-triangular Cholesky factor of C_λ. Re-parametrising in terms of the whitened parameters $\boldsymbol{\xi}$ and $\boldsymbol{\zeta}$, results in the joint posterior

$$\begin{aligned}&\pi(\boldsymbol{\zeta}, \boldsymbol{\xi}, \lambda, \sigma_\varepsilon^2 \mid \mathbf{y}) \\ &\quad \propto \mathcal{N}(\mathbf{y} \mid AL(R_\lambda \boldsymbol{\zeta} + \boldsymbol{\mu}_\ell)^{-1}\boldsymbol{\xi}, \sigma_\varepsilon^2 I_m)\mathcal{N}(\boldsymbol{\xi} \mid 0, I_n)\mathcal{N}(\boldsymbol{\zeta} \mid 0, I_n)\pi(\lambda)\pi(\sigma_\varepsilon^2).\end{aligned}$$

The sampling method is described in Supplementary Algorithm S.2. As opposed to the MWG, the log length scales $\mathbf{u}$ are updated jointly through the whitened parameter $\boldsymbol{\zeta}$. In this case, the likelihood can be evaluated as a product of univariate Gaussian distributions, after computing $\mathbf{u} = R_\lambda \boldsymbol{\zeta} + \boldsymbol{\mu}_\ell$ and solving $L(\mathbf{u})\mathbf{z} = \boldsymbol{\xi}$. Regardless of the hyperprior employed, the latter system of equations $L(\mathbf{u})\mathbf{z} = \boldsymbol{\xi}$ can be solved in $O(n)$ operations by taking advantage of the tridiagonal structure of $L(\mathbf{u})$ [46]. The former system of equations $\mathbf{u} = R_\lambda \boldsymbol{\zeta} + \boldsymbol{\mu}_\ell$ requires matrix multiplication, resulting in $O(n^2)$ operations; however, for the AR(1) hyperprior, we can equivalently solve $L(\lambda)(\mathbf{u} - \boldsymbol{\mu}_\ell) = \boldsymbol{\zeta}$ and make use of the banded form of $L(\lambda)$ to reduce this to $O(n)$ operations.

Thus, while MWG requires looping over the elements of the n-dimensional log length-scale vector, with each operation costing $O(1)$ operations for the AR(1) hyperprior and $O(n)$ operations for the SE hyperprior, the w-ELL-SS instead updates this vector jointly through $O(n)$ for the AR(1) hyperprior and $O(n^2)$ operations for the SE hyperprior. However, as ELL-SS is a rejection free sampling method, each iteration may require several likelihood evaluations, mitigating any gain in computation time of this scheme.

4.3. Marginal elliptical slice sampling (m-ELL-SS)

In simulation studies, we found that integrating out the unknown function $\mathbf{z}$ significantly improves the mixing of $\mathbf{u}$ and its hyperparameters. The log

marginal likelihood of the data corresponds to

$$\log \pi(\mathbf{y} \mid \mathbf{u}, \lambda, \sigma_\varepsilon^2) = -\frac{m}{2}\log(2\pi) - \frac{1}{2}\log\det(\Psi) - \frac{1}{2}\mathbf{y}^{\mathrm{T}}\Psi^{-1}\mathbf{y}, \tag{4.1}$$

where $\Psi = AQ_{\mathbf{u}}^{-1}A^{\mathrm{T}} + \sigma_\varepsilon^2 I_m$. Again, we use whitening to decouple $\mathbf{u}$ and λ, with the re-parametrisation $\boldsymbol{\zeta} = R_\lambda^{-1}(\mathbf{u} - \boldsymbol{\mu}_\ell)$ and $R_\lambda = L(\lambda)^{-1}$ for the AR(1) hyperprior or $R_\lambda = \mathrm{chol}(C_\lambda)$ for the SE hyperprior. The posterior is

$$\pi(\boldsymbol{\zeta}, \lambda, \sigma_\varepsilon^2 \mid \mathbf{y}) \propto \mathcal{N}(\mathbf{y} \mid 0, AQ_{R_\lambda\boldsymbol{\zeta}+\boldsymbol{\mu}_\ell}^{-1}A^{\mathrm{T}} + \sigma_\varepsilon^2 I_n)\mathcal{N}(\boldsymbol{\zeta} \mid 0, I_m)\pi(\lambda)\pi(\sigma_\varepsilon^2).$$

The sampling scheme is detailed in Supplementary Algorithm S.3. Again, the log length scales $\mathbf{u}$ are updated jointly through the whitened parameter $\boldsymbol{\zeta}$. This requires first computing $\mathbf{u} = R_\lambda\boldsymbol{\zeta} + \boldsymbol{\mu}_\ell$, an $O(n)$ operation for the AR(1) hyperprior and $O(n^2)$ operation for the SE hyperprior. However, in comparison with the w-ELL-SS, which proceeds by solving $L(\mathbf{u})\mathbf{z} = \boldsymbol{\xi}$ and simply taking the product of univariate Gaussians in $O(n)$ operations, we must evaluate the marginal likelihood in (4.1).

When computing the marginal likelihood, we emphasise that the required calculations for Ψ can be computed employing the Woodbury identity;

$$\Psi^{-1} = \sigma_\varepsilon^{-2}\left(I_m - \sigma_\varepsilon^{-2}A\left(L(\mathbf{u})^{\mathrm{T}}L(\mathbf{u}) + \sigma_\varepsilon^{-2}A^{\mathrm{T}}A\right)^{-1}A^{\mathrm{T}}\right).$$

While this identity also requires a matrix inversion, note that $L(\mathbf{u})^{\mathrm{T}}L(\mathbf{u}) + \sigma_\varepsilon^{-2}A^{\mathrm{T}}A$ is also banded and therefore computations are considerably cheaper. Indeed, the quadratic term in the marginal likelihood (4.1) is

$$\sigma_\varepsilon^{-2}\left(\mathbf{y}^{\mathrm{T}}\mathbf{y} - \sigma_\varepsilon^{-2}\mathbf{y}^{\mathrm{T}}A\left(L(\mathbf{u})^{\mathrm{T}}L(\mathbf{u}) + \sigma_\varepsilon^{-2}A^{\mathrm{T}}A\right)^{-1}A^{\mathrm{T}}\mathbf{y}\right),$$

with the most expensive operation of order $O(n)$. Specifically, the first term $\mathbf{y}^{\mathrm{T}}\mathbf{y}$ can be computed in $O(m)$ operations, while the second term can be efficiently computed by breaking it into three separate operations. First, we set $\boldsymbol{\varsigma} = A^{\mathrm{T}}\mathbf{y}$, with computational complexity reduced from $O(nm)$ to $O(n)$ through sparsity in A. Next, we solve $(L(\mathbf{u})^{\mathrm{T}}L(\mathbf{u}) + \sigma_\varepsilon^{-2}A^{\mathrm{T}}A)\boldsymbol{\varrho} = \boldsymbol{\varsigma}$ in $O(n)$ operations due to the banded form of the matrix. Finally, we compute $\boldsymbol{\varsigma}^{\mathrm{T}}\boldsymbol{\varrho}$, with a cost of $O(n)$ operations. Computing the determinant, on the other hand, is more expensive with the dominant term costing $O(m^3)$ or $O(nm)$, whichever is greater. Specifically, we must first solve $(L(\mathbf{u})^{\mathrm{T}}L(\mathbf{u}) + \sigma_\varepsilon^{-2}A^{\mathrm{T}}A)B = A^{\mathrm{T}}$, with complexity $O(nm)$, and then compute AB, with reduced complexity $O(nm)$ due to sparsity in A. Finally, the determinant of the $m \times m$ matrix Ψ^{-1} is computed.

In addition, when proposing new values for the noise variance σ_ε^2 or the length scale λ, we must recompute the marginal likelihood (4.1), as opposed to evaluating the product of m univariate Gaussians for the w-ELL-SS scheme, increasing the cost of these steps as well. However, in the marginal scheme, in contrast to both MWG and w-ELL-SS, sampling of $\mathbf{z}$ is no longer required. We also note the computational gains of the AR(1) over the SE hyperprior deteriorate when the determinant evaluation dominates this computation, i.e. when $m^3 > n^2$.

The increased computational cost of the marginal scheme comes with improved mixing, and this trade-off is examined in the simulation studies of Section 6.3. In contrast to MWG, this scheme performs well regardless of the hyperprior employed. Differently from Paciorek and Schervish [39], who also integrate out the non-stationary process and utilise a Metropolis-Hastings step to sample what they called the eigenprocess, we use elliptical slice sampling. This last difference is key in the performance of the proposed algorithm.

5. Extensions for d-dimensional input spaces

To extend the model from Section 3 to higher dimensional input spaces, while maintaining its computational benefits, a novel construction is proposed utilising additive Gaussian process models [AGP, 12]. First, the model is presented, followed by a description of the extended inference procedure.

5.1. Sparse non-stationary additive models

Additive regression models decompose the regression function into main effects and interactions. Linear regression is a classic example, and nonparametric additive models [16, 4] provide increased flexibility, while retaining interpretability and robustness to the input dimension, when compared with general nonparameteric surfaces. The additive GP formulation results from considering the sum and product of covariance functions, two operations for constructing valid covariance functions in d-dimensions. This provides a flexible and interpretable model for the unknown function to include main first-order terms up to d-order interaction terms, assumed to be separable across dimensions.

In the additive GP, the choice between low-order and high-order terms represents a trade-off between between interpretability and accuracy. On one hand, by including only first-order terms, the model can capture long-range structures and has increased intrepretability. On the other, including only a d-order separable function increases flexibility and complexity. Duvenaud et al. [12] include all interaction terms and develop a maximum marginal likelihood approach to determine the importance of each term. Additionally, they develop an efficient algorithm, despite the exponential number of terms, through parametrisations that limit the number of hyperparameters. Interestingly, their experiments show that typically only a few orders of interactions are important. Alternatively, the choice of terms in the additive GP may be application driven; more recently, this is the approach taken in Cheng et al. [6] for longitudinal biomedical data. Another interesting direction in Gilboa et al. [19] constructs projected additive GPs through first-order functions of linear projections of the inputs.

For notational simplicity, in the following, we focus on the 2-dimensional input setting, including both the main and interaction terms for generality. The model construction and inference can be applied to d-dimensional input settings, through appropriate choice of the terms to include in the additive formulation. In two-dimensional input problems, the discretisation is based on a complete

$n_1 \times n_2$ grid, with the noisy realisations modelled through

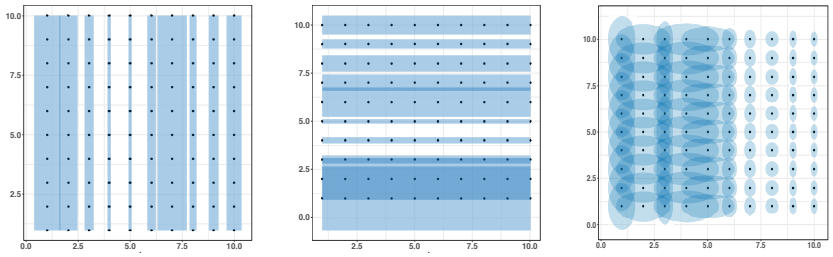
$$\mathbf{y} = A_1\mathbf{z}_1 + A_2\mathbf{z}_2 + A_3\mathbf{z}_3 + \boldsymbol{\varepsilon},$$

where $A_1 \in \mathbb{R}^{m\times n_1}$, $A_2 \in \mathbb{R}^{m\times n_2}$ and $A_3 \in \mathbb{R}^{m\times(n_1 n_2)}$ are known matrices. We assume $z_1(\cdot)$ and $z_2(\cdot)$ are independent one-dimensional non-stationary processes, while $z_3(\cdot)$ is a two-dimensional, separable non-stationary process. Thus, $\mathbf{z}_r \in \mathbb{R}^{n_r}$ denotes the vector formed by the first-order non-stationary processes at the n_r locations in dimension $r = 1, 2$, while $\mathbf{z}_3 \in \mathbb{R}^{n_1 n_2}$ collects the second-order non-stationary process at all locations on the complete $n_1 \times n_2$ grid.

The hierarchical structure of the model (depicted in Figure 3) is

$$\begin{aligned}
\mathbf{y} \mid \{\mathbf{z}_r\}_{r=1}^3, \sigma_\varepsilon^2 &\sim \mathcal{N}(A_1\mathbf{z}_1 + A_2\mathbf{z}_2 + A_3\mathbf{z}_3, \sigma_\varepsilon^2 I_m),\\
\mathbf{z}_r \mid \mathbf{u}_r &\sim \mathcal{N}\left(0, C^{\mathrm{NS}}_{\mathbf{u}_r}\right), \quad r = 1, 2,\\
\mathbf{z}_3 \mid \mathbf{u}_3, \mathbf{u}_4 &\sim \mathcal{N}\left(0, C^{\mathrm{NS}}_{\mathbf{u}_3,\mathbf{u}_4}\right),\\
\mathbf{u}_s \mid \lambda_s &\sim \mathcal{N}\left(\boldsymbol{\mu}_{\ell_s}, C^{\mathrm{S}}_{\lambda_s}\right), \quad s = 1, 2, 3, 4,\\
(\sigma_\varepsilon^2, \boldsymbol{\lambda}) &\sim \pi(\sigma_\varepsilon^2)\pi(\lambda_1)\pi(\lambda_2)\pi(\lambda_3)\pi(\lambda_4),
\end{aligned} \tag{5.1}$$

with $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_4)$. In equation (5.1), we have four one-dimensional length-scale processes: two describing the correlation changes in each direction independently and two incorporating that information in a two-dimensional process, through a separable assumption $C^{\mathrm{NS}}_{\mathbf{u}_3,\mathbf{u}_4}(\mathbf{x}_i, \mathbf{x}_j) = C^{\mathrm{NS}}_{\mathbf{u}_3}(x_{i,1}, x_{j,1})C^{\mathrm{NS}}_{\mathbf{u}_4}(x_{i,2}, x_{j,2})$. A visualisation of the non-stationary additive covariance function is provided in Figure 2.



(a) 1st-order non-stationary covariance $C^{\mathrm{NS}}_{\mathbf{u}_1}$

(b) 1st-order non-stationary covariance $C^{\mathrm{NS}}_{\mathbf{u}_2}$

(c) 2nd-order non-stationary covariance $C^{\mathrm{NS}}_{\mathbf{u}_3,\mathbf{u}_4}$

Figure 2: The non-stationary additive covariance function in 2-d with main effects and an interaction is the sum of the three terms: C^{NS}=$C^{\mathrm{NS}}_{\mathbf{u}_1}$ +$C^{\mathrm{NS}}_{\mathbf{u}_2}$ +$C^{\mathrm{NS}}_{\mathbf{u}_3,\mathbf{u}_4}$. At each location the covariance function will make use the data contained within the shaded region in each of the plots. The 1st-order terms can pool together data across dimensions for long-range correlations, while the 2nd-order terms can capture local behavior in both dimensions.

Because the AGP is based on one-dimensional kernels, we can directly apply the methodology discussed in Section 3 for any of the hyperpriors studied. Instead, a direct extension of the SPDE model to two-dimensional input settings

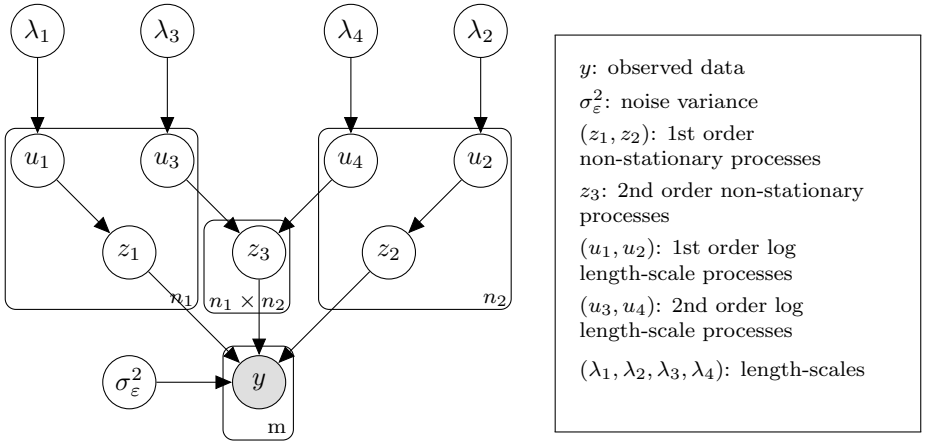


Figure 3: Plate diagram for a non-stationary 2-level additive GP model.

will not allow us to employ the AR(1) hyperprior and benefit from its computational advantages. This is because a two-dimensional exponential covariance does not have a valid Markov representation. Additionally, The SPDE (2.8) depends on the input dimensions, specifically it is assumed that $\nu = 2 - d/2$, this means that for a 3-dimensional process we will recover the non-stationary exponential covariance, which for most real-world applications will be too rough to be a realistic assumption. Finally, the additive and hierarchical structure of the model in equation (5.1) favours interpretability about the behaviour of the correlation in each dimension.

5.2. Inference for additive non-stationary models

The posterior for the additive non-stationary model in equation (5.1) is

$$\begin{aligned}\pi(\{\mathbf{z}_r\}_{r=1}^3, \{\mathbf{u}_s, \lambda_s\}_{s=1}^4, \sigma^2_\varepsilon \mid \mathbf{y}) \propto \mathcal{N}(\mathbf{y} \mid A_1\mathbf{z}_1 + A_2\mathbf{z}_2 + A_3\mathbf{z}_3, \sigma^2_\varepsilon I_m) \\ \mathcal{N}(\mathbf{z}_1 \mid 0, Q^{-1}_{\mathbf{u}_1})\mathcal{N}(\mathbf{z}_2 \mid 0, Q^{-1}_{\mathbf{u}_2})\mathcal{N}(\mathbf{z}_3 \mid 0, Q^{-1}_{\mathbf{u}_3,\mathbf{u}_4}) \\ \mathcal{N}(\mathbf{u}_1 \mid \boldsymbol{\mu}_{\ell_1}, C_{\lambda_1}) \cdots \mathcal{N}(\mathbf{u}_4 \mid \boldsymbol{\mu}_{\ell_4}, C_{\lambda_4})\pi(\lambda_1) \cdots \pi(\lambda_4)\pi(\sigma^2_\varepsilon),\end{aligned}$$

with $Q^{-1}_{\mathbf{u}_3,\mathbf{u}_4}$ being a separable covariance matrix, defined as $Q^{-1}_{\mathbf{u}_{3,4}} := Q^{-1}_{\mathbf{u}_3} \otimes Q^{-1}_{\mathbf{u}_4}$, where $\otimes$ denotes the Kronecker product. The three inference schemes described in Section 4 can be appropriately extended through a blocked Gibbs sampler, that updates the three blocks of parameters $(\mathbf{z}_1, \mathbf{u}_1, \lambda_1)$; $(\mathbf{z}_2, \mathbf{u}_2, \lambda_2)$; and $(\mathbf{z}_3, \mathbf{u}_3, \mathbf{u}_4, \lambda_3, \lambda_4)$ from their full conditional distributions. Following from the one-dimensional synthetic experiments of Section 6.1, we focus on the marginal sampler of Section 4.3. We will refer to it as the block marginal elliptical slice sampler (Block-m-ELL-SS); in this case, although we are not integrating out the processes $\{\mathbf{z}_r\}_{r=1}^3$, we use the marginal likelihood to sample the length-scale process and corresponding length-scale hyperparameters in each block. For instance, when sampling the block $(\mathbf{z}_1, \mathbf{u}_1, \lambda_1)$, the full conditional factorises as

$$\pi(\mathbf{z}_1, \boldsymbol{\zeta}_1, \lambda_1 \mid \mathbf{y}, \sigma^2_\varepsilon, \mathbf{z}_2, \mathbf{z}_3) = \pi(\boldsymbol{\zeta}_1, \lambda_1 \mid \mathbf{y}, \sigma^2_\varepsilon, \mathbf{z}_2, \mathbf{z}_3)\pi(\mathbf{z}_1 \mid \boldsymbol{\zeta}_1, \lambda_1, \mathbf{y}, \sigma^2_\varepsilon, \mathbf{z}_2, \mathbf{z}_3),$$

with $\boldsymbol{\zeta}_1 = R_{\lambda_1}^{-1}(\mathbf{u}_1 - \boldsymbol{\mu}_{\ell_1})$ denoting the whitened parameter. Thus, we first sample from the block marginal $\pi(\boldsymbol{\zeta}_1, \lambda_1 \mid \mathbf{y}, \sigma_\varepsilon^2, \mathbf{z}_2, \mathbf{z}_3)$ utilising the steps described in Section 4.3, with the marginal likelihood replaced by $\mathcal{N}(\mathbf{y} - A_2\mathbf{z}_2 - A_3\mathbf{z}_3|0, A_1Q_{\mathbf{u}_1}^{-1}A_1^{\mathrm{T}} + \sigma_\varepsilon^2 I_m)$. The algorithm is detailed in Supplementary Algorithm S.4. For efficiency in evaluating the block marginal likelihood obtained from integration of $\mathbf{z}_r$, $r = 1, 2$, the matrix determinant lemma [23] must be employed to avoid computing the determinant of an $m \times m$ matrix and instead evaluate the determinant of three small matrices.

We note that for a d-dimensional problem with n_k data points in dimension $k = 1, 2, \ldots, d$ and where only first-order terms are included in the model, the algorithm requires $\mathcal{O}(\sum_{k=1}^{d} n_k m)$ operations per marginal likelihood evaluation in order to sample the reparametrised length-scale process and its corresponding length-scale hyperparameters. In addition, due to whitening one needs $\mathcal{O}(\sum_{k=1}^{d} n_k)$ operations for the AR hyperprior compared to $\mathcal{O}(\sum_{k=1}^{d} n_k^2)$ under the SE hypermodel.

When an interaction term is employed in the model, the algorithm requires samples from the posterior of $\mathbf{z}_3$, which is a Gaussian distribution with mean $\boldsymbol{\mu}_{z_3} = \sigma_\varepsilon^{-2}\Sigma_{z_3}A_3^{\mathrm{T}}(\mathbf{y} - A_1\mathbf{z}_1 - A_2\mathbf{z}_2)$ and variance $\Sigma_{z_3} = (Q_{\mathbf{u}_3} \otimes Q_{\mathbf{u}_4} + \sigma_\varepsilon^{-2}A_3^{\mathrm{T}}A_3)^{-1}$. These posterior moment computations need the inversion of an $n_1n_2 \times n_1n_2$ matrix and cannot exploit the Kronecker structure because of the second summand in Σ_{z_3}. To overcome this, we utilise the efficient method of Gilboa et al. [19, Section 2.2], based on eigendecompositions and matrix-vector multiplications for Kronecker matrices. This procedure applies to the case when $A_3^{\mathrm{T}}A_3 = I_{n_1n_2}$; this constraint requires the data to be observed on the complete grid (not necessarily equidistant), but can easily be relaxed for incomplete grids and domain extensions with an additional Gibbs step to sample the missing observations. Specifically, we make use of the identity

$$\begin{aligned}\Sigma_{z_3} &= \left(Q_{\mathbf{u}_3} \otimes Q_{\mathbf{u}_4} + \sigma_\varepsilon^{-2} I_{n_1n_2}\right)^{-1} \\ &= E_3 \otimes E_4(\Lambda_3 \otimes \Lambda_4 + \sigma_\varepsilon^{-2} I_{n_1n_2})^{-1} E_3^{\mathrm{T}} \otimes E_4^{\mathrm{T}},\end{aligned} \tag{5.2}$$

where $Q_{\mathbf{u}_3} = E_3\Lambda_3E_3^{\mathrm{T}}$ and $Q_{\mathbf{u}_4} = E_4\Lambda_4E_4^{\mathrm{T}}$, with E_3 and E_4 denoting the eigenvectors matrices and Λ_3 and Λ_4 denoting the diagonal matrices of eigenvalues of $Q_{\mathbf{u}_3}$ and $Q_{\mathbf{u}_4}$, respectively. Exploiting the sparse structure in the precision matrices permits to reduce the cubically computatitional complexity of the required eigen-decompositions in this step. The second key identity is

$$(E_3 \otimes E_4)\boldsymbol{\alpha} = \mathrm{vec}[(E_3[E_4\,\mathrm{reshape}(\boldsymbol{\alpha}, n_2, n_1)]^{\mathrm{T}})^{\mathrm{T}}], \tag{5.3}$$

where the operator $\mathrm{reshape}(b, p, q)$ returns a $p \times q$ matrix whose elements are taken from the vector b, and $\mathrm{vec}(M)$ denotes the vectorisation of a matrix M. Importantly, (5.2) and (5.3) permit to reduce the number of operatios from $\mathcal{O}((n_1n_2)^3)$ to $\mathcal{O}(n_1n_2)$.

Thus, to efficiently compute the posterior mean, $\boldsymbol{\mu}_{z_3}$, we follow three steps:

$$\begin{aligned}\boldsymbol{\alpha} &= \operatorname{vec}\left[(E_3^{\mathrm{T}}[E_4^{\mathrm{T}}\operatorname{reshape}(\tilde{\mathbf{y}}, n_2, n_1)]^{\mathrm{T}})^{\mathrm{T}}\right],\\ \boldsymbol{\alpha} &= (\Lambda_3 \otimes \Lambda_4 + \sigma_\varepsilon^{-2} I_{n_1 n_2})^{-1}\boldsymbol{\alpha},\\ \boldsymbol{\mu}_{z_3} &= \sigma_\varepsilon^{-2}\operatorname{vec}\left[(E_3[E_4\operatorname{reshape}(\boldsymbol{\alpha}, n_2, n_1)]^{\mathrm{T}})^{\mathrm{T}}\right],\end{aligned}$$

where $\tilde{\mathbf{y}} := \mathbf{y} - A_1\mathbf{z}_1 - A_2\mathbf{z}_2$. Note that $(\Lambda_3 \otimes \Lambda_4 + \sigma_\varepsilon^{-2} I_{n_1 n_2})$ is diagonal and therefore easy to invert. A posterior sample of $\mathbf{z}_3$ is then obtained by sampling $\boldsymbol{\eta} \sim \mathcal{N}(0, I_{n_1 n_2})$ and setting $\mathbf{z}_3 = \boldsymbol{\mu}_{z_3} + E_3 \otimes E_4(\Lambda_3 \otimes \Lambda_4 + \sigma_\varepsilon^{-2} I_{n_1 n_2})^{-1/2}\boldsymbol{\eta}$, where for the latter operation, we again make use of the second identity (5.3) and the diagonal form of $(\Lambda_3 \otimes \Lambda_4 + \sigma_\varepsilon^{-2} I_{n_1 n_2})$. The last critical computation is the evaluation of the block marginal likelihood $\mathcal{N}(\tilde{\mathbf{y}} \mid 0, Q_{\mathbf{u}_3}^{-1} \otimes Q_{\mathbf{u}_4}^{-1} + \sigma_\varepsilon^2 I_{n_1 n_2})$, which is required to sample $(\boldsymbol{\zeta}_3, \boldsymbol{\zeta}_4)$ and the corresponding hyperparameters, λ_3 and λ_4. First, the quadratic term can be calculated efficiently following the approach employed for the posterior mean. Next, for the log determinant computation, one can use again the eigendecomposition; namely,

$$\begin{aligned}&\log\det\left(Q_{\mathbf{u}_3}^{-1} \otimes Q_{\mathbf{u}_4}^{-1} + \sigma_\varepsilon^2 I_{n_1 n_2}\right)^{-1}\\ &\quad= \log\det\left(E_3 \otimes E_4(\Lambda_3^{-1} \otimes \Lambda_4^{-1} + \sigma_\varepsilon^2 I_{n_1 n_2})^{-1} E_3^{\mathrm{T}} \otimes E_4^{\mathrm{T}}\right)\\ &\quad= -\log\det\left(\Lambda_3^{-1} \otimes \Lambda_4^{-1} + \sigma_\varepsilon^2 I_{n_1 n_2}\right),\end{aligned}$$

where $\Lambda_3^{-1} \otimes \Lambda_4^{-1} + \sigma_\varepsilon^2 I_{n_1 n_2}$ is a diagonal matrix, whose log determinant is straightforward to calculate, reducing the computational completity of the log determinant computation from $\mathcal{O}((n_1 n_2)^3)$ to $\mathcal{O}(n_1 n_2)$. We emphasize that the required terms can also be efficiently computed for higher-order interactions through d-dimensional versions of the two key identities (5.2) and (5.3) in Gilboa et al. [19]. Consequently, the computational cost of including d-order interaction terms is $\mathcal{O}(\prod_{k=1}^{d} n_k)$.

6. Experiments

We apply the sparse non-stationary hierarchical methodology to three simulated one-dimensional interpolation experiments and a two-dimensional synthetic example. First, the one-dimensional experiments study the effects of the discretisation and sample size on the efficiency of the algorithms presented in Section 4 under two extreme hyperpriors. In addition, the experiments show that our model can recover different signal types, while also providing information on the correlation structure. Second, a two-dimensional synthetic experiment demonstrates how the model can be extended to higher dimensional input spaces utilising an AGP model. Finally, in Section 6.3, we present a comparative evaluation on the performance of 2-level GP models against two other methods: a stationary GP model and a Bayesian treed GP [TGP, 20] model, a popular approach for dealing with non-stationarity.

6.1. One-dimensional synthetic data

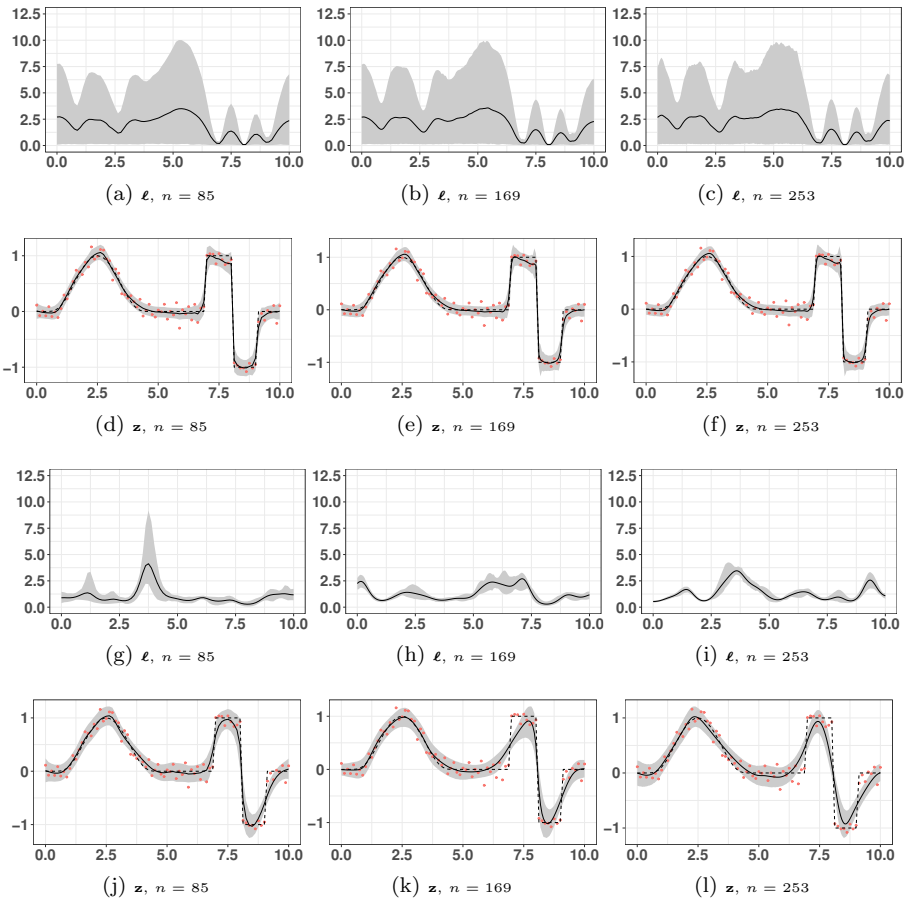
We consider three simulated datasets with different signal types. The first example (Supplementary Figure S.1a) is a function with smooth parts and edges and is also piecewise constant. The second synthetic dataset (Supplementary Figure S.1b) is a damped sine wave function with smooth decaying oscillations. The third example corresponds to the *Bumps* (Supplementary Figure S.1c) function employed by Donoho and Johnstone [8], which depicts a signal with pronounced spikes and constant parts. In the first dataset, we investigate, empirically, posterior consistency of the estimates with respect to the discretisation scheme. The second experiment explores the performance of the sampling schemes for increased sample size and measurement noise. The last example examines emphasises the importance of the prior choice.

Experiment 1: Smooth-piecewise constant function

For all experiments, we use the same initialisation and run the chains for $T = 200,000$ iterations. The burn-in period is algorithm specific, selected according to preliminary runs based on Raftery and Lewis's diagnostic [41] for the second level length-scale. Numerical discretisation-invariance is studied by varying n in the experiments, with $n = 85, 169, 253$. The mean and variance of the prior length-scale process is set at zero and one, respectively. For the second level length-scale, we use a broad prior, $\log \lambda \sim \mathcal{N}(0, 3)$.

We start by presenting the results obtained with the MWG algorithm. Figure 4 shows estimates of the spatially varying length-scales and the unknown function under both hyperpriors. For the AR(1) hyperprior, an inspection of traceplots and cumulative averages of the estimates (not shown) suggest convergence of the chains for all discretisation schemes. In addition, the varying length-scale estimates exhibit the expected behaviour (i.e. decaying when the function has a sharp jump and increasing when the function is constant), and the interpolated estimates indicate a reasonable fit to the unknown function for all three discretisations schemes (Figure 4(a)-(f)). However, this is not the case for the SE hyperprior. Figure 4(g)-(l) illustrates the results obtained with this hyperprior for the same sampling algorithm. Under this setting, the effect of discretisation scheme is evident. As we increase n, the method fails to recover the unknown function. The strong correlation between the elements of $\mathbf{u}$ induced by the SE hyperprior makes the algorithm converge rather slowly to the target distribution.

In contrast to the results obtained with MWG, both w-ELL-SS and m-ELL-SS demonstrate convergence for both hyperpriors and invariance to the discretisation (see Supplementary Figures S.2 and S.3 for a complete analysis). Figure 5 summarises succinctly important differences in mixing across the algorithms by showing traceplots with cumulative averages for a subset of parameters. The results are shown for the most challenging scenario, SE hyperprior at the highest resolution, $n = 253$. Figure 5(a)(d) emphasises the lack of convergence for MWG. Figure 5(b)(e) demonstrates the high autocorrelation of the chains and the slow convergence produced by w-ELL-SS. Finally, Figure 5(c)(f) highlights



(a) $\boldsymbol{\ell}$, $n = 85$ (b) $\boldsymbol{\ell}$, $n = 169$ (c) $\boldsymbol{\ell}$, $n = 253$

(d) $\mathbf{z}$, $n = 85$ (e) $\mathbf{z}$, $n = 169$ (f) $\mathbf{z}$, $n = 253$

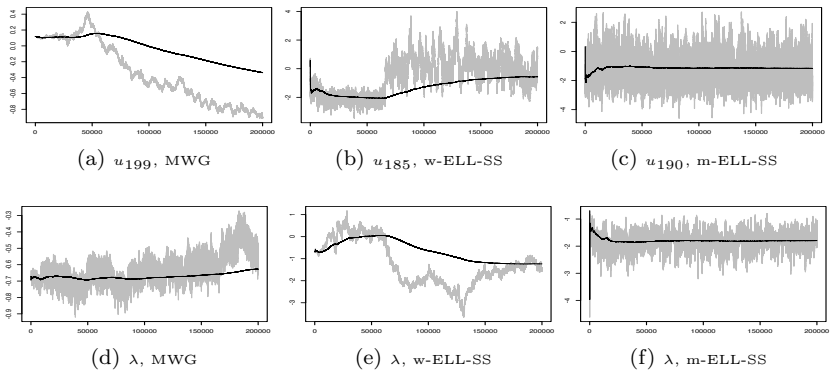
(g) $\boldsymbol{\ell}$, $n = 85$ (h) $\boldsymbol{\ell}$, $n = 169$ (i) $\boldsymbol{\ell}$, $n = 253$

(j) $\mathbf{z}$, $n = 85$ (k) $\mathbf{z}$, $n = 169$ (l) $\mathbf{z}$, $n = 253$

Figure 4: Results for Experiment 1 with MWG. (a)-(c): Estimated $\boldsymbol{\ell}$ process with 95% credible intervals for AR(1) hyperprior on different grids. (d)-(f): Estimated $\mathbf{z}$ process with 95% credible intervals for AR(1) hyperprior on different grids with observed data in red. (g)-(i): Estimated $\boldsymbol{\ell}$ process with 95% credible intervals for SE hyperprior on different grids. (j)-(l): Estimated $\mathbf{z}$ process with 95% credible intervals for SE hyperprior on different grids with observed data in red.

the improvement offered by m-ELL-SS, fast convergence to the stationary distribution and low autocorrelation of the chains.

In order to evaluate the performance of the algorithms, we show in Table 1 an overall efficiency score (OES) of the chains [51]. This measure considers both the CPU time (Supplementary Table S.2) required to run the chains and the effective sample size (ESS) (Supplementary Table S.3). The score is com-



(a) u_{199}, MWG (b) u_{185}, w-ELL-SS (c) u_{190}, m-ELL-SS

(d) λ, MWG (e) λ, w-ELL-SS (f) λ, m-ELL-SS

Figure 5: Traceplots with cumulative averages of the chains for SE hyperprior with $n = 253$. (Top row:) element of $\mathbf{u}$ with the lowest ESS. (Bottom row:) the hyperparameter.

puted as OES = ESS/CPUtime[2]. For both multidimensional vectors, $\mathbf{z}$ and $\mathbf{u}$, we report the OES computed with the minimum ESS across all dimensions. The results indicate that while MWG with the AR(1) hyperprior shows high efficiency for some parameters when $n = 85$, its performance deteriorates as n increases. This suggests that this sampling scheme will not perform efficiently for bigger datasets even when $m = n$ (this is explored in Experiment 2). Furthermore, despite the fact that MWG reports the lowest CPU time under the AR(1) hyperprior (Supplementary Table S.2), its overall efficiency scores are outperformed by those obtained with m-ELL-SS; this is due to the low autocorrelation of the chains achieved by the marginal sampler (see Supplementary Table S.3). In contrast, chains of the parameters for w-ELL-SS result in the worse OES. Notice also that the scores reported for MWG with the SE hyperprior are not informative as the chains show convergence problems. Table 1 also reports mean absolute error (MAE) to evaluate the fit to the unknown function and the empirical coverage of the 95% credible intervals (EC) to evaluate accuracy in uncertainty quantification. For the SE hyperprior, w-ELL-SS and m-ELL-SS report equivalent errors and EC, while MWG yields worse values.

Experiment 2: Damped sine wave

This example explores the effect of increasing the sample size and measurement noise. Due to robustness of the estimates with respect to the discretisation in the first example, we only present experiments for the discretisation scheme when $m = n$. The chains are run for $T = 100,000$ iterations with a burn-in period that is algorithm and prior specific. In addition, we extend the domain with 40 points on each side of the interval, such that $n = 430$ and $m = 350$. The prior distributions for $\mathbf{u}$ and $\log \lambda$ are as in Experiment 1.

[2]All experiments were run in an Intel Core i7-6700 CPU (3.40GHz, 16 GB of RAM).

		MWG			w-ELL-SS			m-ELL-SS		
		$n = 85$	$n = 169$	$n = 253$	$n = 85$	$n = 169$	$n = 253$	$n = 85$	$n = 169$	$n = 253$
AR(1)	σ^2_ε	622.76	173.12	65.99	380.89	102.38	38.91	**661.20**	**257.81**	**116.35**
	ℓ_{min}	**635.36**	114.02	41.05	30.90	8.99	2.94	287.16	**114.36**	**59.71**
	z_{min}	**203.80**	42.10	13.91	9.12	2.34	0.86	129.75	**52.16**	**22.30**
	λ	89.84	15.66	6.00	22.77	5.26	2.36	**111.80**	**45.54**	**21.53**
	MAE	0.041	0.051	0.054	0.041	0.051	0.054	0.041	0.051	0.053
	EC	0.988	0.975	0.971	0.988	0.975	0.975	0.988	0.975	0.975
SE	σ^2_ε	11.19	4.88	7.49	246.24	77.72	8.89	**856.15**	**253.91**	**125.97**
	ℓ_{min}	1.22	0.73	0.64	21.69	10.22	2.79	**244.91**	**122.57**	**55.82**
	z	0.06	0.01	0.01	4.71	1.37	0.24	**76.80**	**24.11**	**9.87**
	λ	0.59	0.75	0.31	2.31	0.29	0.01	**16.59**	**4.15**	**2.21**
	MAE	0.078	0.100	0.133	0.040	0.050	0.054	0.039	0.049	0.052
	EC	0.889	0.826	0.763	0.988	0.975	0.971	0.988	0.975	0.979

Table 1: Experiment 1: OES with both hyperpriors under various discretisation schemes ($n = 86, 169, 253$) and three different algorithms. ℓ_{min} and z_{min} report OES for the minimum ESS across all dimensions. Highest values in boldface.

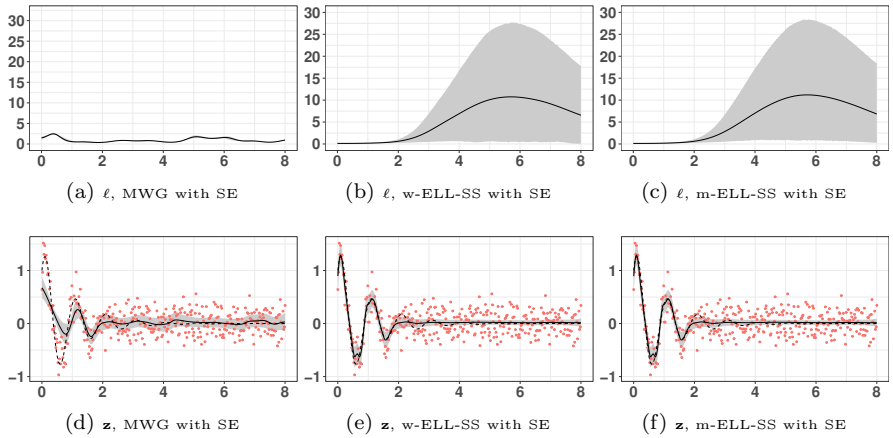


Figure 6: Results for Experiment 2. Top row: estimated $\boldsymbol{\ell}$ process with 95% credible interval for SE hyperprior with (a) MWG, (b) w-ELL-SS and (c) m-ELL-SS. Second row: estimated $\mathbf{z}$ process with 95% credible interval for SE hyperprior with (d) MWG, (e) w-ELL-SS and (f) m-ELL-SS.

While the results with the AR(1) hyperprior appear satisfactory under the three sampling schemes (Supplementary Figure S.4), once again, SE hyperprior (Figure 6) with MWG is not able to explore the posterior of $\mathbf{u}$, resulting in poor estimates and hence, the highest MAE and poor EC (see Table 2). Analysing the efficiency of the samplers, first, for the AR hyperprior, we observe that while MWG is faster (Table S.4), its ESS is consistently smaller (Supplementary Table S.6), hence reducing its OES (Table 2). In contrast to the findings in Experiment 1, w-ELL-SS reports better OES compared to MWG due to better mixing in

	AR(1)			SE		
	MWG	w-ELL-SS	m-ELL-SS	MWG	w-ELL-SS	m-ELL-SS
σ_ε^2	12.73	**27.54**	14.21	0.27	**32.29**	15.27
ℓ_{min}	0.06	0.14	**0.65**	0.00	0.40	**1.04**
z_{min}	0.13	0.13	**0.75**	0.01	0.55	**1.41**
λ	0.19	0.36	**0.95**	0.02	0.05	**0.25**
MAE	0.038	0.039	0.039	0.089	0.038	0.038
EC	0.920	0.934	0.934	0.863	0.940	0.934

Table 2: Experiment 2: OES with AR(1) and SE hyperprior employing three different algorithms. ℓ_{min} and z_{min} report OES for the minimum ESS across all dimensions. Highest values in boldface.

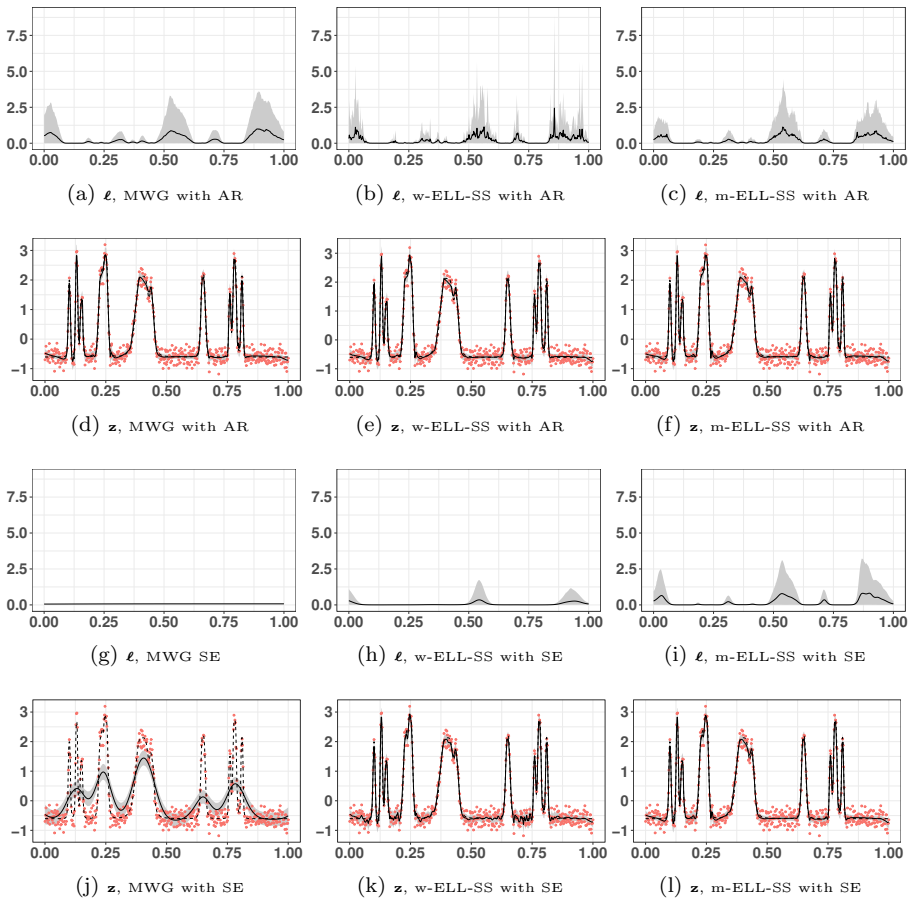
the chains. We believe this is due to the noise level, which favours a whitened parametrisation. Finally, despite the fact that the marginal sampler reports larger CPU times, the low correlation of its chains (Supplementary Table S.6) favours its OES. Second, when using the SE hyperprior, the marginal sampler appears to be significantly faster and consistently reports the best OES. This, together with the negligible differences in MAE and EC, suggests that m-ELL-SS offers a good compromise between computational cost and efficiency, with the benefit of working well under highly correlated priors.

Experiment 3: Bumps

The data is generated employing the *Bumps* function in Donoho and Johnstone [8] and scaled to have zero mean and unit variance. Following Vannucci and Corradi [52], we generate $m = 512$ points in the interval [0,1] and use a signal-to-noise ratio equal to 5, such that $\sigma_\varepsilon^2 = .04$. To avoid a boundary problem, we extend the domain with 30 points on each side of the interval, such that $n = 572$. Chains are run for $T = 100,000$ iterations with algorithm and prior specific burn-in periods. We use empirical priors for the log length-scale process and log length-scale hyperparameter; namely, $\mu_\ell = -3.06$, $\tau_\ell^2 = 2.62$, and $\log \lambda \sim \mathcal{N}(-3.06, 2.62)$ (see Supplementary Section 4.3.1 for more details on prior elicitation).

This example highlights important differences between the two hyperpriors and the proposed MCMC algorithms. First, under the AR(1) hyperprior, the three sampling schemes show differences in the posterior length-scale process (Figure 7(a)-(c)). While MWG results in a smooth process, m-ELL-SS and w-ELL-SS appear to be more sensitive to the prior, with rougher estimates. Second, for the SE hyperprior, once more, MWG did not reach convergence. Also, the performance of w-ELL-SS has become impaired; the posterior length-scale process does not reflect the changes in the correlation structure, and the length-scale hyperparameter did not reach the stationary distribution. The posterior length-scale process obtained with m-ELL-SS appears more appropriate, although, still shows a prior effect.

The findings discussed above are also evidenced in the OES shown in Table 3, where MWG exhibits the highest scores and the lowest MAE under AR(1). In contrast, the m-ELL-SS scheme outperforms MWG and w-ELL-SS for a SE



(a) $\boldsymbol{\ell}$, MWG with AR (b) $\boldsymbol{\ell}$, w-ELL-SS with AR (c) $\boldsymbol{\ell}$, m-ELL-SS with AR

(d) $\mathbf{z}$, MWG with AR (e) $\mathbf{z}$, w-ELL-SS with AR (f) $\mathbf{z}$, m-ELL-SS with AR

(g) $\boldsymbol{\ell}$, MWG SE (h) $\boldsymbol{\ell}$, w-ELL-SS with SE (i) $\boldsymbol{\ell}$, m-ELL-SS with SE

(j) $\mathbf{z}$, MWG with SE (k) $\mathbf{z}$, w-ELL-SS with SE (l) $\mathbf{z}$, m-ELL-SS with SE

Figure 7: Results for Experiment 3. Top row: estimated $\boldsymbol{\ell}$ process with 95% credible interval for AR(1) hyperprior with (a) MWG, (b) w-ELL-SS and (c) m-ELL-SS. Second row: estimated $\mathbf{z}$ process with 95% credible interval for AR(1) hyperprior with (d) MWG, (e) w-ELL-SS and (f) m-ELL-SS. Third row: estimated $\boldsymbol{\ell}$ process with 95% credible interval for SE hyperprior with (g) MWG, (h) w-ELL-SS and (i) m-ELL-SS. Bottom row: estimated $\mathbf{z}$ process with 95% credible interval for SE hyperprior with (j) MWG, (k) w-ELL-SS and (l) m-ELL-SS.

hyperprior. We believe the differences illustrated in this experiment are a result of a key challenge of elliptical slice sampling. When the likelihood is strong, the sampler can result in poor mixing and, in extreme cases, can get stuck [13]. In addition, when sampling kernel parameters in strong likelihood settings, one can expect a non-centred parametrisation (avoiding whitening) to be more efficient (see Section 3 in Murray and Adams [36]).

The computational time required for this experiment is reported in Supplementary Table S.9. Given the same initial values, the marginal sampler converges to the stationary distribution faster; indeed, m-ELL-SS reports, across

	AR(1)			SE		
	MWG	w-ELL-SS	m-ELL-SS	MWG	w-ELL-SS	m-ELL-SS
σ^2_ε	**23.42**	5.73	5.70	2.06	5.48	**15.36**
ℓ_{min}	0.01	0.01	**0.13**	0.00	0.01	**0.15**
z_{min}	**2.43**	0.10	0.24	0.56	0.07	**0.85**
λ	**0.65**	0.03	0.13	**0.07**	0.00	0.03
MAE	0.060	0.061	0.062	0.461	0.069	0.060
EC	0.955	0.950	0.959	0.385	0.961	0.967

Table 3: Experiment 3: OES with AR(1) and SE hyperprior employing three different algorithms. ℓ_{min} and z_{min} report OES for the minimum ESS across all dimensions. Highest values in boldface.

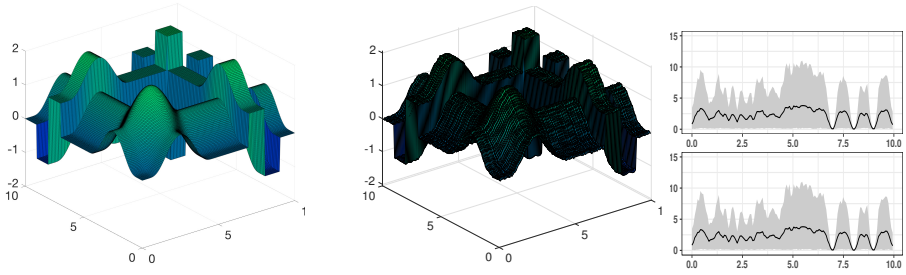
experiments, the smallest time spent in burn-in period. Finally, to highlight how the model can benefit from using a more powerful computer, we ran this experiment in an Intel Xeon E5-260V3 2.4GHz (Haswell), 8-core processors with 32 GB of RAM, and we found that the inference procedure is sped up by a factor of ≈ 2.1 for m-ELL-SS and w-ELL-SS (see Supplementary Table S.10). However, for MWG, the speed up factor was only ≈ 1.2.

6.2. d-dimensional synthetic datasets

We demonstrate the performance of our approach on two synthetic datasets with $d = 2$ and 3. Firstly, for the 2-d dataset (Experiment 4) we generate $m = 20,449$ noisy observations in an expanded grid of $n_1 = n_2 = 143$ equally spaced points in $\big[0, 10\big]$, employing $z(x_1, x_2) = z(x_1) + z(x_2)$, where both $z(x_1)$ and $z(x_2)$ correspond to the function used in Experiment 1. The noise variance is set to $\sigma^2_\varepsilon = 0.06$ and we use the same prior distributions of Experiment 1 for each of the length-scale processes and corresponding hyperparameters. Secondly, Experiment 5 corresponds to a 3-d dataset with $m = 592,704$ data points generated in an equally spaced 3 dimensional grid in $\big[0, 10\big]$ with $z(x_1, x_2, x_3) = z(x_1) + z(x_2) + z(x_3)$, where $z(x_1)$ uses the function of Experiment 1, $z(x_2) = \sin(x_2/2)$, and $z_3 = -\exp(-3(x_3 - 2.5)^2)\mathbb{1}_{x_3<5} + \exp(-3(x_3 - 7.5)^2)\mathbb{1}_{x_3\geq 5}$. The noise variance is set to 0.02 and we use empirical priors for the length-scale processes and its hyperparameters. In both cases, the samplers were run for $T = 50,000$ iterations, with a burn-in of $10,000$.

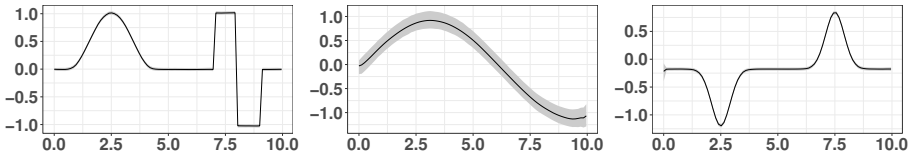
Figure 8 shows the results for the 2-d example, the figure depicts the true surface versus the posterior mean obtained from a 2-level AGP model (without interaction term), employing the Block-m-ELL-SS algorithm. Our model is able to capture the smooth areas and edges of the surface. In addition, it provides information about the correlation structure along each axis (Figure 8(b)). The 2-level AGP correctly learns the varying correlation along the surface; for instance, the true function in the region $[5, 6] \times [5, 6]$ is constant, and in the same region, the 1-d length-scale processes depict strong correlation. The required total computational time for this experiment was 99.26 minutes (19.67 in burn-in and 79.59 in non-burned).

Figure 9 illustrates the results for our 3-dimensional synthetic example. Figures 9(a)-(c) show posterior estimates of the one-dimensional non-stationary

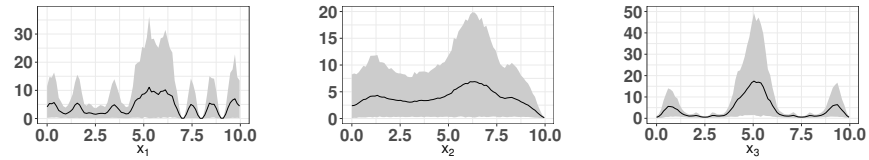


(a) True surface (b) Fitted surface and length-scale processes

Figure 8: Results for 2-dimensional synthetic data. (a): True surface. (b): Posterior mean surface and one-dimensional length-scale processes with 95% credible intervals.



(a) z_1 (b) z_2 (c) z_3



(d) ℓ_1 (e) ℓ_2 (f) ℓ_3

Figure 9: Results for 3-dimensional synthetic data. (a)-(c): Posterior mean for the one-dimensional non-stationary processes with 95% credible intervals (d)-(e): Posterior estimate for the stationary length-scales processes with 95% credible intervals.

processes showing that the method is able to recover the true generating functions. Figures 9(d)-(f) exhibit the posterior length-scale processes in each dimension, illustrating that our approach permits to learn differences in the correlation structure. In this case, the inference proceedure took 927.6 minutes.

6.3. Comparative evaluation

We offer a comparative evaluation of our model for the synthetic examples from Section 6.1 and 6.2. We compare against: 1) stationary Mátern Gaussian process (STAT) with $\nu = 1.5$, 2) Bayesian treed Gaussian process (TGP) [20], 3) local approximate Gaussian process (LAGP) [21], and 4) generalized additive models (GAM) [54].

For the stationary model, the length scale and noise variance are inferred via MCMC, employing a marginal sampler with adaptive random walks. The GP prior mean and magnitude are fixed at 0 and 1, respectively, as in the 2-level GP model. For the TGP, we consider a stationary Matern kernel with $\nu = 1.5$ and a constant mean function. The magnitude is also inferred, in contrast to the stationary and the 2-level model. In order to make use of the default prior distributions, we rescale the response and inputs, as recommended by the authors. For LAGP we do a grid search over the parameters and report the results with the lowest MAE. Similarly, for GAM we use cubic penalized regression splines and run a grid search for the free parameters in the model. Results of the grid search for LAGP and GAM are available in the Supplementary material, Section 5.2 and 5.3, respectively.

For STAT, TGP and 2-level GP, the chains for all the experiments are run for the same number of iterations $(100,000)$, with the same burnin period $(20,000)$. In addition, for STAT and 2-level GP the chains are initialised with the same values. For our d-dimensional simulated dataset (Experiment 4 and 5), it is computationally unfeasible to run STAT and TGP models due to the size of the datasets. Instead, to offer a comparison, we consider a subset of the original 2-dimensional dataset, reducing the data size from $20,449$ to 441 observations (Experiment 4 subset).

Figure 10 shows the posterior mean estimates of the unknown under the three models for the three different 1-d synthetic datasets, and Figure 11 illustrates the posterior mean surface for the subset of data in Experiment 4. Note that the grey areas depict the 95% credible intervals of the unknown function for STAT and 2-level GP but, instead, depict the 95% credible intervals of the noisy observations for TGP. This is because storing region-specific traces is memory intensive, and the storage is not supported in the tgp package without doing predictions. In addition, Table 4 and 5 report MAE and EC of the experiments, where we report EC of the noisy process for TGP.

		Experiment 1 ($m = 81$)	Experiment 2 ($m = 350$)	Experiment 3 ($m = 512$)
STAT	MAE	0.076	0.047	0.094
	EC	0.914	0.946	0.947
TGP	MAE	0.056	0.043	0.079
	EC	0.963	0.934	0.963
LAGP	MAE	0.072	0.091	0.111
	EC	NA	NA	NA
GAM	MAE	0.083	0.048	0.089
	EC	NA	NA	NA
2 LEVEL (AR/SE)	MAE	0.041/**0.039**	0.039/**0.038**	0.062/**0.060**
	EC	0.988/0.988	0.934/0.940	0.959/0.967

Table 4: Comparative evaluation on 1-dimensional synthetic datasets. For Experiments 1-3 with 2-level GP model, we employ m-ELL-SS algorithm for both hyperpriors. Best values in boldface.

First, the results make clear the downside of applying a stationary model

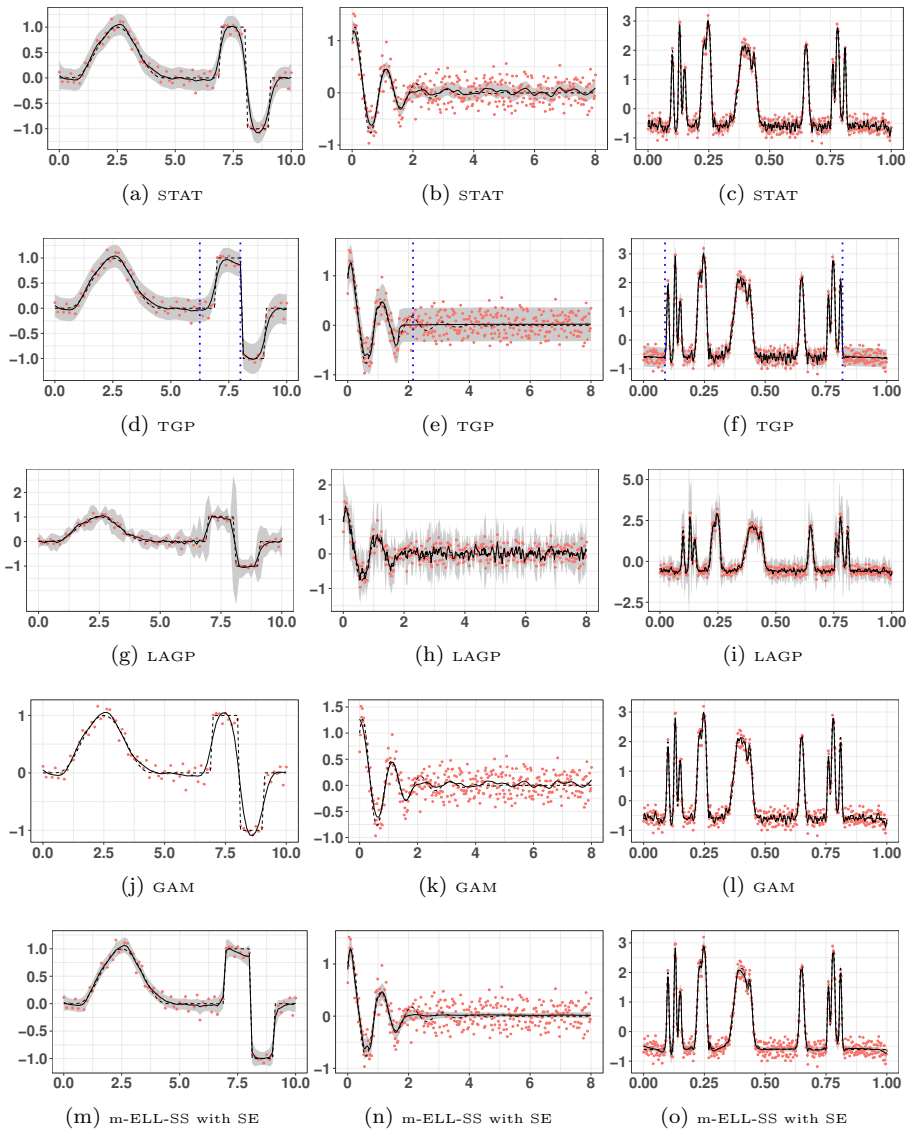
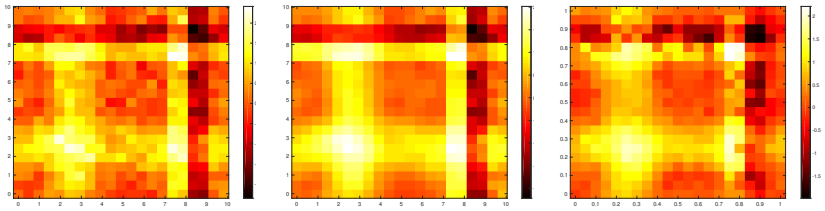
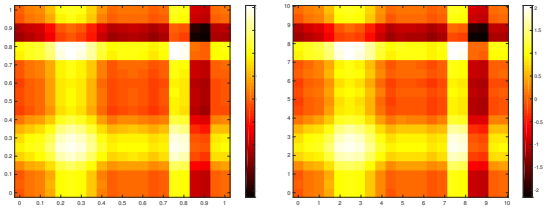


Figure 10: Comparative evaluation for 1-d experiments. Each column shows one of the simulated experiments. Red dots depict observed data, dotted lines show the true signal, solid lines show the posterior mean, and grey areas depict 95% credible intervals. (a)-(c): Stationary GP (d)-(f): TGP, with blue dotted lines depicting MAP cut-off points. (g)-(i): LAGP with lowest MAE, (j)-(l): GAM with lowest MAE, and (m)-(o): 2-level GP with m-ELL-SS algorithm and the hyperprior with lowest MAE.

to non-stationary data in all four compared experiments. In Experiment 1,



(a) STAT (b) TGP (c) LAGP



(d) GAM (e) 2-level GP

Figure 11: Comparative evaluation for 2-d experiment. Posterior mean surface for (a): anisotropic stationary model, (b): TGP, (c): LAGP with lowest MAE, (d): GAM with first order terms and lowest MAE, and (e): 2-level AGP with first order terms.

		Experiment 4-subset ($m = 441$)	Experiment 4 ($m = 20,449$)
STAT	MAE	0.195	NA
	EC	0.501	NA
TGP	MAE	0.122	NA
	EC	0.980	NA
LAGP	MAE	0.213	0.091
	EC	NA	NA
GAM	MAE	**0.071**	0.030
	EC	NA	NA
2 LEVEL (AR)	MAE	0.072	**0.020**
	EC	0.963	0.959

Table 5: Comparative evaluation on the 2-dimensional synthetic datasets. Experiment 4 on the full dataset is computationally unfeasible for STAT and TGP. For Experiment 4 with $m = 20,449$ LAGP and GAM report results based on the grid search obtained with the subset. The 2-level AGP model uses block-m-ELL-SS with AR hyperprior. EC* for TGP is reported for the noisy process. Best values in boldface.

STAT is oversmoothing and unable to capture the edges in the function (see Figure 10(a)). Example 2 and 3 (Figures 10(b)(c)) illustrate how a stationary model tends to overfit when the function is constant, as a result of the different characteristics of the unknown. The same behaviour is repeated in the two-dimensional synthetic example (Figure 11(a)).

Second, while TGP offers an improvement, compared with a stationary set-

ting, the model still oversmooths where the function possesses an edge. For instance, in Figure 10(d), the partition found around 6.2 is misplaced, and a third partition should be included around 9 to capture correctly the edges. In Experiment 2 (Figure 10(e)), the partition is also misplaced; this is however more reasonable (compared to Experiment 1) due to the smooth change in the behaviour. In Experiment 3, despite the fact that TGP fit is good when the function is constant (Figure 10(f)), the main limitation appears to be in finding some of the partitions that are required to ameliorate the issues resulting from fitting piecewise stationary models. Note that we ran TGP with a different number of iterations ($100,000$; $200,000$ and $500,000$) to verify the results shown in Figure 10 and 11 (see Supplementary Section 5 for the results). In Experiment 3, while increasing the number of iterations has a positive effect on the partitions found (and therefore on MAE), it was not enough to outperform the 2-level GP model. Also, this was not the case for the other experiments, where increasing the number of iterations either did not affect the fit or worsened it. Moreover, without knowing the ground truth, it would be hard to know beforehand if the algorithm has been run for long enough to find the appropriate partitions.

Third, LAGP appear to overfit in all three one-dimensional datasets (Figure 10(g)-(i)) and in the subset of Experiment 4 (Figure 11(c)). We believe this is due to the small data size of these experiments that do not permit smooth transitions. Note that when we use the full dataset in Experiment 4, the MAE considerably drops (see Table 5) compared to the results obtained when using a subset of the data. This issue is further discuss in Gramacy [21, Section 3.3].

Finally, for our one-dimensional experiments, GAM shows the same limitations than the stationary GP, as it considerably oversmooths in the edges and overfits when the function is constant (see Figure 10(j)-(l)). As illustrated in Figure 11(d) GAM seems competitive in the subset of our 2-dimensional dataset, producing very similar results to those obtain with 2-level AGP (Figure 11(e)). However, when we increase the datasize in Experiment 4 the performance of GAM deteriorates (see Table 5).

In summary, the 2-level GP is an alternative model for non-stationary data that resolves the issues discussed above. It does not overfit or oversmooth, works well in small and big dataset, and it appears to be more efficient in dealing with different types of non-stationarities, such as, edges, smooth changes or sharp peaks. Moreover, the 2-level GP clearly benefits from the additive structure, making the model scalable, while retaining flexibility. Notice that evaluating the methods solely on running time can be misleading, as STAT and 2-level GP are implemented in R using standard libraries, while TGP uses R as front end to call `C` and `C++` optimised code, and LAGP permits parallelisation in several manners.

6.4. Real data: NASA rocket booster vehicle

The analysed dataset in this experiment comes from a computer simulator of a NASA rocket booster vehicle, the Langley Glide-Back Booster [22]. NASA scientists are interested in understanding the behaviour of the rocket when it re-enters the atmosphere. To do so, the computer experiment considers six

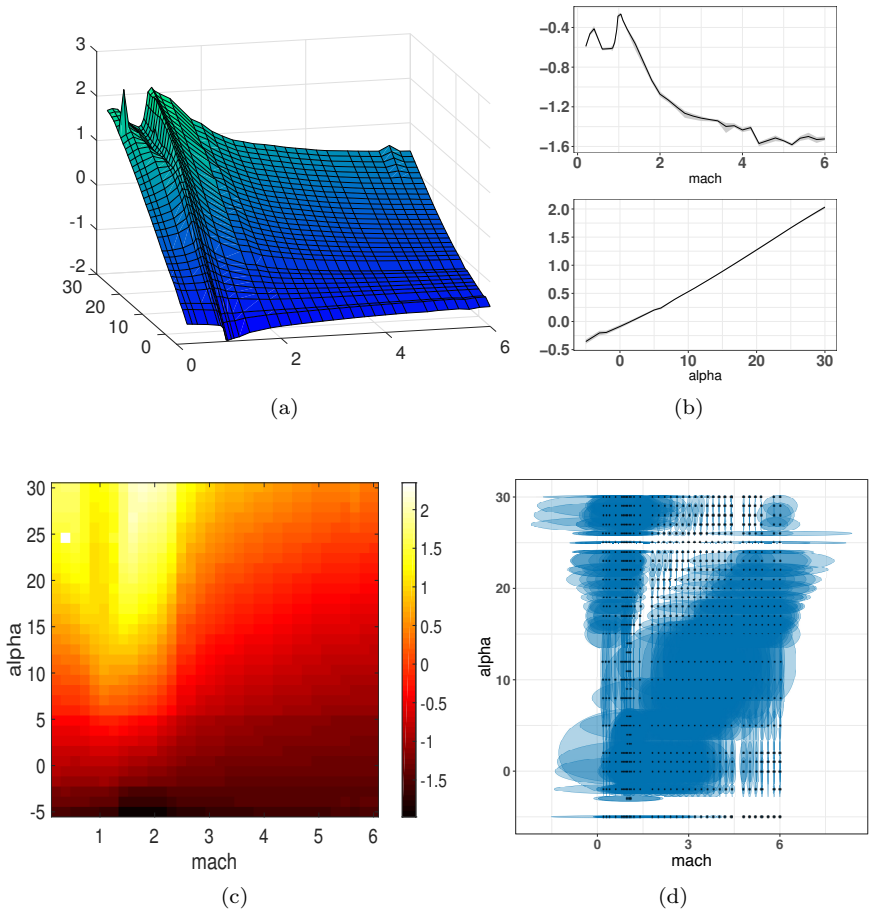


Figure 12: Results for NASA rocket booster vehicle. (a): Posterior mean of the non-stationary process, (b): Posterior mean of the two one-dimensional processes with 95% credible intervals, (c): Heatmap of the posterior mean depicted in (a), (d): Covariance structure for the 2nd-order term.

different variables; lift, drag, pitch, side force, yaw, and roll; all forces that keep the rocket up. Here, we focus on how the lift force is affected as a function of the speed (mach) and the angle of attack (alpha) for a particular value of the slide-slip angle (beta=0). The data is, by nature, non-stationary, with different levels of smoothness along the surface and with a ridge showing the change from subsonic to supersonic flow at mach=1 and large alpha.

The data consists on 861 observations on a 34×33 grid where the speed ranges from $[0.2, 6]$ and the angle of attack from $[-5, 30]$. The data is more dense for mach values around one. Thus, the data is available on an incomplete, non-equally spaced, rectangular grid. We consider the 2-level AGP model with

interaction term, employing the Block-m-ELL-SS algorithm for inference. In order to deal with missing values, we use the model to impute them at each iteration of the MCMC. The chain is run for 50,000 iterations with a burn-in period of 10,000.

Figure 12(a) shows the posterior mean obtained. The model is able to capture the expected ridge around mach=1 and a sharp peak in the boundary around alpha=25, where the latter seems to be an error in the convergence of the simulator [22]. Furthermore, Figure 12(b) illustrates the posterior mean of each of the one-dimensional processes. The results suggest that fitting a stationary process for the angle of attack (alpha) may be enough. Importantly, Figure 12(d) illustrates how the 2nd-order term can capture local behavior of the process in both dimensions. Depictions of the posterior mean of the second-order interaction term and all length-scale processes are provided in the Supplementary Material. The required computational time for this experiment was 5.78 hours in a high performance cluster.

Because in this case we do not have the true function, to provide a comparison with TGP, we split the available data at random, using $m = 461$ observations for the training set and $m^* = 200$ as test set. In this case, the performance in terms of computational time and predicitve errors of both methods is similar. Nevertheless, we expect to see greater benefits for larger dataset. Specifically, our model reports a lower error in terms of MSE = 0.001 compared to TGP with a MSE = 0.002; however, in terms of MAE, TGP appear to do better with a MAE = 0.021, in contrast to a MAE = 0.030 for our model. This difference, higlights that when abrupt changes are important our 2-level GP model is more efficient. Supplementary Figure S.13 provides scatter plots of true versus predicted under the two models.

7. Discussion

We constructed non-stationary hierarchical models based on stochastic parameters and Gaussian Markov random fields, ameliorating the computational constraints of doing exact inference in 2-level GP models through sparsity in the finite-dimensional approximation of the inverse covariance matrix of the non-stationary field. Different hyperpriors were also explored for the spatially varying length-scale, from strong prior smoothness assumptions through a squared exponential covariance to rough hyperpriors of an autoregressive AR(1) model, with the latter benefiting from further computational gains. Strong dependence between the model layers makes efficient inference challenging, and to address this, we introduced and investigated the performance of three different MCMC algorithms. First, we found that the Metropolis-within-Gibbs scheme performs poorly for highly correlated hyperpriors and exhibits deteriorating efficiency as the number of observations or discretisation size increase. Second, the whitened elliptical slice sampler performs well for weak likelihoods, regardless the hyperprior employed, at the price of highly correlated chains. Finally, the marginal elliptical slice sampler appears to be an efficient strategy to break the correla-

tion between latent process and hyperparameters and offers a good compromise between computational complexity and efficiency of the chains.

We also proposed a novel extension to d-dimensional settings by combining additive Gaussian process models with 2-level GPs. The additive structure and use of Kronecker algebra for the interaction term result in an inference procedure that is tractable and scalable. Our experiments show that the additive structure retains the flexibility of the 2-level GP and favours its interpretability. Moreover, while we focus on the two-dimensional setting, the additive 2-level model and inference scheme naturally extend to higher dimensions. Overall, the comparative evaluation highlights the benefits of our approach, over stationary and popular non-stationary GP models, to recover edges, peaks and smooth variations in the data in both one-dimensional and two-dimensional settings. In addition, the methodology may benefit greatly from using powerful computational resources.

The experiments presented here suggest that the algorithms based on elliptical slice sampling do not deteriorate as the resolution becomes finer or the sample size increases, similar to the schemes discussed by Chen et al. [5]. However, it is important to emphasise that elliptical slice sampling is known to perform well for weak data likelihoods; therefore, care must be taken in the small noise limit. Furthermore, it would be interesting to explore the performance of the auxiliary gradient-based sampling scheme recently proposed by Titsias and Papaspiliopoulos [51]; however, notice that this scheme requires derivatives, which for our model are expensive and not straightforward to compute. We also highlight the recent work of Durrande et al. [10], implementing banded matrix operators in TensorFlow, which, combined with GPflow [34], could provide a promising direction for automatic differentiation for our model.

While this work exemplifies the methodology on a computer simulation problem, the approach here discussed is applicable for a range of data. In addition, a natural extension of this work is to the 3-level GP model or, more generally, the deep GP models studied in Dunlop et al. [9]. Other interesting directions for future research include exploring higher-order autoregressive hyperpriors; more general kernels; and alternative likelihoods for problems beyond regression, such as the classification and inverse problems discussed in Chen et al. [5].

Acknowledgements

The work reported in this paper was funded by the Mexican National Council of Science and Technology (CONACYT) grant no. CVU609843; the Engineering and Physical Sciences Research Council, grant no. EP/K034154/1; and the Academy of Finland, grant nos. 326240 and 326341, and with support from the Alan Turing Institute - Lloyds Register Foundation programme on data-centric engineering.

Appendix A. Supplementary Material

Supplementary Material associated with this article can be found online : `http://www.some-url-address.com`, and code will be made publicly available.

References

[1] Anderes, E. B., Stein, M. L., 2008. Estimating deformations of isotropic Gaussian random fields on the plane. The Annals of Statistics 36 (2), 719–741.

[2] Berrocal, V. J., Raftery, A. E., Gneiting, T., Steed, R. C., 2010. Probabilistic weather forecasting for winter road maintenance. Journal of the American Statistical Association 105 (490), 522–537.

[3] Blomqvist, K., Kaski, S., Heinonen, M., 2018. Deep convolutional Gaussian processes. arXiv preprint arXiv:1810.03052.

[4] Buja, A., Hastie, T., Tibshirani, R., 1989. Linear smoothers and additive models. The Annals of Statistics 17 (2), 453–510.

[5] Chen, V., Dunlop, M. M., Papaspiliopoulos, O., Stuart, A. M., 2019. Dimension-robust MCMC in Bayesian inverse problems. arXiv preprint arXiv:1803.03344.

[6] Cheng, L., Ramchandran, S., Vatanen, T., Lietzn, N., Lahesmaa, R., Vehtari, A., Lähdesmäki, H., 2019. An additive Gaussian process regression model for interpretable non-parametric analysis of longitudinal data. Nature Communications 10 (1798).

[7] Damianou, A., Lawrence, N., 2013. Deep Gaussian processes. In: Artificial Intelligence and Statistics. pp. 207–215.

[8] Donoho, D. L., Johnstone, I. M., 1995. Adapting to unknown smoothness via wavelet shrinkage. Journal of the American Statistical Association 90 (432), 1200–1224.

[9] Dunlop, M. M., Girolami, M., Stuart, A. M., Teckentrup, A. L., 2018. How deep are deep Gaussian processes? Journal of Machine Learning Research 19, 1–46.

[10] Durrande, N., Adam, V., Bordeaux, L., Eleftheriadis, S., Hensman, J., 2019. Banded matrix operators for Gaussian Markov models in the automatic differentiation era. In: Artifical Intelligence and Statistics.

[11] Duvenaud, D., Rippel, O., Adams, R., Ghahramani, Z., 2014. Avoiding pathologies in very deep networks. In: Artificial Intelligence and Statistics. pp. 202–210.

[12] Duvenaud, D. K., Nickisch, H., Rasmussen, C. E., 2011. Additive Gaussian processes. In: Advances in Neural Information Processing Systems. pp. 226–234.

[13] Fagan, F., Bhandari, J., Cunningham, J., 2016. Elliptical slice sampling with expectation propagation. In: Uncertainty in Artificial Intelligence.

[14] Filippone, M., Zhong, M., Girolami, M., 2013. A comparative evaluation of stochastic-based inference methods for Gaussian process models. Machine Learning 93 (1), 93–114.

[15] Fouedjio, F., Desassis, N., Rivoirard, J., 2016. A generalized convolution model and estimation for non-stationary random functions. Spatial Statistics 16, 35–52.

[16] Friedman, J. H., Stuetzle, W., 1981. Projection pursuit regression. Journal of the American Statistical Association 76 (376), 817–823.

[17] Fuglstad, G. A., Lindgren, F., Simpson, D., Rue, H., 2015. Exploring a new class of non-stationary spatial Gaussian random fields with varying local anisotropy. Statistica Sinica 25 (1), 115–133.

[18] Fuglstad, G. A., Simpson, D., Lindgren, F., Rue, H., 2015. Does non-stationary spatial data always require non-stationary random fields? Spatial Statistics 14, 505–531.

[19] Gilboa, E., Saatçi, Y., Cunningham, J. P., 2015. Scaling multidimensional inference for structured Gaussian processes. IEEE Transactions on Pattern Analysis and Machine Intelligence 37 (2), 424–436.

[20] Gramacy, R. B., 2007. tgp: an R package for Bayesian nonstationary, semiparametric nonlinear regression and design by treed Gaussian process models. Journal of Statistical Software 19 (9), 1–46.

[21] Gramacy, R. B., 2016. laGP: Large-scale spatial modeling via local approximate gaussian processes in R. Journal of Statistical Software 72 (1), 1–46.

[22] Gramacy, R. B., Lee, H. K., 2008. Bayesian treed Gaussian process models with an application to computer modeling. Journal of the American Statistical Association 103 (483), 1119–1130.

[23] Harville, D. A., 1997. Matrix algebra from a statistician's perspective. Vol. 1. Springer.

[24] Heaton, M. J., Datta, A., Finley, A. O., Furrer, R., Guinness, J., Guhaniyogi, R., Gerber, F., Gramacy, R. B., Hammerling, D., Katzfuss, M., Lindgren, F., Nychka, D. W., Sun, F., Zammit-Mangion, A., Dec 2018. A case study competition among methods for analyzing large spatial data. Journal of Agricultural, Biological and Environmental Statistics.

[25] Hegde, P., Heinonen, M., Lähdesmäki, H., Kaski, S., 2019. Deep learning with differential Gaussian process flows. In: Artificial Intelligence and Statistics. Vol. 89. pp. 1812–1821.

[26] Heinonen, M., Mannerström, H., Rousu, J., Kaski, S., Lähdesmäki, H., 2016. Non-stationary Gaussian process regression with Hamiltonian Monte Carlo. In: Artificial Intelligence and Statistics. pp. 732–740.

[27] Kaipio, J., Somersalo, E., 2006. Statistical and computational inverse problems. Springer Science & Business Media.

[28] Katzfuss, M., 2013. Bayesian nonstationary spatial modeling for very large datasets. Environmetrics 24 (3), 189–200.

[29] Kim, H.-M., Mallick, B. K., Holmes, C., 2005. Analyzing nonstationary spatial data using piecewise Gaussian processes. Journal of the American Statistical Association 100 (470), 653–668.

[30] Kleiber, W., Nychka, D., 2012. Nonstationary modeling for multivariate spatial processes. Journal of Multivariate Analysis 112, 76–91.

[31] Lang, T., Plagemann, C., Burgard, W., 2007. Adaptive non-stationary kernel regression for terrain modeling. In: Robotics: Science and Systems.

[32] Lindgren, F., Rue, H., Lindström, J., 9 2011. An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. Journal of the Royal Statistical Society: Series B 73 (4), 423–498.

[33] Matheron, G., 1973. The intrinsic random functions and their applications. Advances in Applied Probability, 439–468.

[34] Matthews, A. G., van der Wilk, M., Nickson, T., Fujii, K., Boukouvalas, A., Leoón-Villagrá, P., Ghahramani, Z., Hensman, J., 2017. GPflow: A Gaussian process library using TensorFlow. Journal of Machine Learning Research 18 (1), 1299–1304.

[35] Montagna, S., Tokdar, S. T., 2016. Computer emulation with nonstationary Gaussian processes. SIAM/ASA Journal on Uncertainty Quantification 4 (1), 26–47.

[36] Murray, I., Adams, R. P., 2010. Slice sampling covariance hyperparameters of latent Gaussian models. In: Advances in Neural Information Processing Systems. pp. 1732–1740.

[37] Murray, I., Adams, R. P., MacKay, D. J., 2010. Elliptical slice sampling. In: Artifical Intelligence and Statistics. Vol. 13. pp. 541–548.

[38] Neto, J. H. V., Schmidt, A. M., Guttorp, P., 2014. Accounting for spatially varying directional effects in spatial covariance structures. Journal of the Royal Statistical Society: Series C 63 (1), 103–122.

[39] Paciorek, C. J., Schervish, M. J., 2006. Spatial modelling using a new class of nonstationary covariance functions. Environmetrics 17 (5), 483–506.

[40] Pintore, A., Holmes, C., 2004. Spatially adaptive non-stationary covariance functions via spatially adaptive spectra. Tech. rep., University of Oxford.

[41] Raftery, A. E., Lewis, S. M., 1992. [Practical Markov Chain Monte Carlo]: Comment: One long run with diagnostics: Implementation strategies for Markov Chain Monte Carlo. Statistical science 7 (4), 493–497.

[42] Risser, M. D., 2016. Nonstationary spatial modeling, with emphasis on process convolution and covariate-driven approaches. arXiv preprint arXiv:1610.02447.

[43] Risser, M. D., Calder, C. A., 2017. Local likelihood estimation for covariance functions with spatially-varying parameters: The convoSPAT package for R. Journal of Statistical Software 81 (1), 1–32.

[44] Roberts, G. O., Rosenthal, J. S., 2009. Examples of adaptive MCMC. Journal of Computational and Graphical Statistics 18 (2), 349–367.

[45] Roininen, L., Girolami, M., Lasanen, S., Markkanen, M., 2019. Hyperpriors for Matérn fields with applications in Bayesian inversion. Inverse Problems and Imaging 13 (1), 1–29.

[46] Rue, H., Held, L., 2005. Gaussian Markov random fields: Theory and applications. Chapman and Hall/CRC.

[47] Sampson, P., Damian, D., Guttorp, P., 2001. Advances in modeling and inference for environmental processes with nonstationary spatial covariance. In: geoENV III — Geostatistics for Environmental Applications. Vol. 11. Springer, pp. 17–32.

[48] Seiler, M. C., Seiler, F. A., 1989. Numerical recipes in C: the art of scientific computing. Risk Analysis 9 (3), 415–416.

[49] Stathopoulos, V., Zamora-Gutierrez, V., Jones, K., Girolami, M., 2014. Bat call identification with Gaussian process multinomial probit regression and a dynamic time warping kernel. In: Artificial Intelligence and Statistics. pp. 913–921.

[50] Stein, M. L., 2005. Nonstationary spatial covariance functions. Tech. rep., Center for Integrating Statistical and Environmental Science, University of Chicago, Chicago.

[51] Titsias, M. K., Papaspiliopoulos, O., 2018. Auxiliary gradient-based sampling algorithms. Journal of the Royal Statistical Society: Series B 80, 749–767.

[52] Vannucci, M., Corradi, F., 1999. Covariance structure of wavelet coefficients: Theory and models in a Bayesian perspective. Journal of the Royal Statistical Society: Series B 61 (4), 971–986.

[53] Volodina, V., Williamson, D. B., 2018. Diagnostic-driven nonstationary emulators using kernel mixtures. arXiv preprint arXiv:1803.04906.

[54] Wood, S., 2019. mgcv: Mixed GAM Computation Vehicle with Automatic Smoothness Estimation. R package version 1.8-31.

[55] Yu, Y., Meng, X.-L., 2011. To center or not to center: That is not the question -An ancillarity-sufficiency interweaving strategy (ASIS) for boosting MCMC efficiency. Journal of Computational and Graphical Statistics 20 (3), 531–570.

[56] Yue, Y. R., Simpson, D., Lindgren, F., Rue, H., 2014. Bayesian adaptive smoothing splines using stochastic differential equations. Bayesian Analysis 9 (2), 397–424.

[57] Zhang, H., 2004. Inconsistent estimation and asymptotically equal interpolations in model-based geostatistics. Journal of the American Statistical Association 99 (465), 250–261.