A flexible Bayesian hierarchical modeling framework for spatially dependent peaks-over-threshold data

Rishikesh Yadav[1], Raphaël Huser[1*] and Thomas Opitz[2]

May 11, 2022

Abstract In this work, we develop a constructive modeling framework for extreme threshold exceedances in repeated observations of spatial fields, based on general product mixtures of random fields possessing light or heavy-tailed margins and various spatial dependence characteristics, which are suitably designed to provide high flexibility in the tail and at sub-asymptotic levels. Our proposed model is akin to a recently proposed Gamma-Gamma model using a ratio of processes with Gamma marginal distributions, but it possesses a higher degree of flexibility in its joint tail structure, capturing strong dependence more easily. We focus on constructions with the following three product factors, whose different roles ensure their statistical identifiability: a heavy-tailed spatially-dependent field, a lighter-tailed spatially-constant field, and another lighter-tailed spatially-independent field. Thanks to the model's hierarchical formulation, inference may be conveniently performed based on Markov chain Monte Carlo methods. We leverage the Metropolis adjusted Langevin algorithm (MALA) with random block proposals for latent variables, as well as the stochastic gradient Langevin dynamics (SGLD) algorithm for hyperparameters, in order to fit our proposed model very efficiently in relatively high spatio-temporal dimensions, while simultaneously censoring non-threshold exceedances and performing spatial prediction at multiple sites. The censoring mechanism is applied to the spatially independent component, such that only univariate cumulative distribution functions have to be evaluated. We explore the theoretical properties of our model, and illustrate the proposed methodology by simulation and application to daily precipitation data from North-Eastern Spain measured at nearly 100 stations over the period 2011–2020.

Keywords: Bayesian hierarchical modeling; extreme event; precipitation; stochastic gradient Langevin dynamics; sub-asymptotic modeling; threshold exceedance.

[1]Computer, Electrical and Mathematical Sciences and Engineering (CEMSE) Division, King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Saudi Arabia. E-mails: rishikesh.yadav@kaust.edu.sa; raphael.huser@kaust.edu.sa (corresponding author)

[2]INRAE, UR546 Biostatistics and Spatial Processes, 228, Route de l'Aérodrome, CS 40509, 84914 Avignon, France. E-mail: thomas.opitz@inra.fr

1 Introduction

Due to its importance for quantifying risk, the statistical modeling of extreme events is crucial in a wide range of environmental applications. Most environmental data are spatial or spatio-temporal in nature, and models at the interface between spatial statistics and Extreme-Value Theory (EVT) provide a mathematically rigorous way to study the marginal behavior of extreme events, while accounting for their potentially strong spatial or spatio-temporal dependence; see, e.g., the review papers by Davison et al. (2012), Cooley et al. (2012), Davison and Huser (2015), Davison et al. (2019). The block maximum (BM) and peaks-over-threshold (POT) approaches are the two principal techniques for modeling the extremes of a probability distribution. Although their pros and cons are still debated (Bücher and Zhou, 2021), the POT approach is usually preferred because of its more natural (and often better) use of available data, its direct modeling of the spatial extreme events that effectively took place, and its ability to model clusters of extreme events. The generalized Pareto (GP) distribution plays a key role in the POT approach, being the only possible limit for the marginal distribution of appropriately rescaled high threshold exceedances; see Davison and Smith (1990).

Various statistical approaches have been proposed for the modeling of spatial extremes, including Bayesian hierarchical models, copula models, max-stable random fields, and generalized Pareto processes; see the review papers by Davison et al. (2012) and Huser and Wadsworth (2020). Recently, Bayesian hierarchical models have gained in popularity for modeling spatial extremes due to their flexibility to capture complex spatio-temporal trends, and their ease of inference in both BM and POT settings, when the data can be assumed to be conditionally independent given some spatially-structured latent variables. Conditional independence is a common assumption in Bayesian hierarchical models (Cressie, 1993; Cooley et al., 2007; Banerjee et al., 2014; Opitz et al., 2018; Johannesson et al., 2021) and drastically simplifies computations, but it also poses a risk of misrepresenting the data-level dependence structure. Based on this assumption, several Bayesian

hierarchical models using latent variables have been proposed in the literature to model high threshold exceedances; see, e.g., Cooley et al. (2007); Opitz et al. (2018); Bacro et al. (2020), and the recent book chapter by Hazra et al. (2021). In particular, Cooley et al. (2007) used Gaussian processes to capture latent spatial dependence and trends in precipitation data, and used a relatively simple Markov chain Monte Carlo (MCMC) algorithm for the estimation of posterior distributions. Similarly, Turkman et al. (2010) fitted Bayesian hierarchical models to spatio-temporal wildfire data from Portugal by taking advantage of MCMC-based inference, while Opitz et al. (2018) and Castro-Camilo et al. (2019) exploited the integrated nested Laplace approximation (INLA) to fit a model to spatio-temporal threshold exceedances. More recently, Hazra et al. (2021) proposed using Max-and-Smooth, an approximate Bayesian algorithm designed for extended latent Gaussian models, and they applied it to a large-scale extreme precipitation dataset.

However, while these hierarchical models typically succeed in estimating marginal distributions accurately, the conditional independence assumption at the data level is very restrictive when the goal is to estimate return levels of spatial aggregates. Conditional independence models indeed yield unrealistic realizations for spatial phenomena such as rainfall or temperature, where threshold exceedances usually produce smooth surfaces (Ribatet et al., 2012). Sang and Gelfand (2010) realized the limitation of the conditional independence model proposed by Sang and Gelfand (2009) and improved it by introducing a Gaussian copula at the data level. Similarly, Clark and Dixon (2021) proposed spatial models based on self-exciting processes to capture strong spatial dependence, allowing both data-level and latent-level dependencies.

As an alternative solution, in this work we extend the hierarchical modeling framework developed by Yadav et al. (2021) to capture relatively strong spatial dependence among threshold exceedances, by mixing several random fields multiplicatively in a way that ensures a heavy-tailed marginal behavior and generates a wide range of joint tail structures. To propose flexible heavy-tailed models, Yadav et al. (2021) relied on Breiman's Lemma (Breiman, 1965), which characterizes

the tail behavior of the product of two nonnegative independent random variables when one of them has power-law tail decay. Let X_1 and X_2 be nonnegative independent random variables such that $\mathbb{E}(X_1^{\alpha+\epsilon}) < \infty$, for some $\epsilon > 0$, and the distribution of X_2 is regularly varying at ∞ with index $-\alpha < 0$, i.e., $\Pr(X_2 > x) = \ell(x)x^{-\alpha}$, where $\ell(x) > 0$ and $\ell(tx)/\ell(t) \to 1$, as $t \to \infty$ (i.e., ℓ is a slowly varying function). Then, we have the tail expansion

$$\Pr(X_1 X_2 > x) \sim \mathbb{E}(X_1^{\alpha}) \Pr(X_2 > x), \qquad x \to \infty. \tag{1}$$

Essentially, the result in (1) implies that the tail decay behavior of the product of two independent random variables, where one is regularly varying and the other one is lighter-tailed, is completely determined by the tail behavior of the regularly varying component, while the lighter-tailed component only contributes a constant scaling factor of tail probabilities. Breiman's Lemma (1) motivates the construction of models with improved flexibility at a sub-asymptotic level, while allowing a heavy-tailed behavior. Yadav et al. (2021) used this result to generalize the GP distribution, which can be obtained from the product of two independent Exponential and Inverse Gamma-distributed random variables. Specifically, they proposed spatial models constructed as $Y(\boldsymbol{s}) = X_1(\boldsymbol{s})X_2(\boldsymbol{s})$, where $X_1(\boldsymbol{s})$ and $X_2(\boldsymbol{s})$ are independent processes that have Gamma and Inverse Gamma margins, respectively, and $\boldsymbol{s} \in \mathbb{R}^2$ is the location in Euclidean space. While the light-tailed Gamma process $X_1(\boldsymbol{s})$ was assumed to be spatial white noise, the heavy-tailed Inverse Gamma process $X_2(\boldsymbol{s})$ was used to incorporate spatial dependence, thus inducing dependence among threshold exceedances. However, because of the spatial independence of $X_1(\boldsymbol{s})$, the range of possible dependence structures that the product $Y(\boldsymbol{s}) = X_1(\boldsymbol{s})X_2(\boldsymbol{s})$ can attain is very limited, restricting the model to relatively weak tail dependencies. This issue can also be seen by reformulating the process $Y(\boldsymbol{s})$ as a Bayesian hierarchical model, whereby $Y(\boldsymbol{s}) \mid X_2(\boldsymbol{s})$ is a conditionally independent Gamma process and $X_2(\boldsymbol{s})$ is a latent spatially-structured random field. The conditional independence assumption at the data level here strongly restricts the form of dependence of $Y(\boldsymbol{s})$. In this paper, we generalize the hierar-

chical spatial model of Yadav et al. (2021) to provide new, more flexible hierarchical spatial models for threshold exceedances, that mitigate the effect of the conditional independence assumption at the data level with the ultimate goal of capturing stronger spatial dependence among extreme events, while retaining the computational benefits and the intuitive interpretation of such Bayesian hierarchical models.

For the spatial modeling of precipitation (for which we usually find heavy tails), we consider instead the product of three suitably defined processes possessing different marginal and dependence characteristics, and with clearly distinct roles in the overall spatial model. More precisely, we assume that our model can be written as

$$Y(\boldsymbol{s}) = \alpha(\boldsymbol{s})X_1(\boldsymbol{s})X_2(\boldsymbol{s})X_3(\boldsymbol{s}), \tag{2}$$

where $X_1(\boldsymbol{s}) \geq 0$ is a unit mean noise process with independent and identically distributed (iid) variables that captures small-scale variations and that allows fast Bayesian computations; $X_2(\boldsymbol{s}) \equiv X_2 \geq 0$ is a fully dependent spatial process with unit mean, which counterbalances the noise process $X_1(\boldsymbol{s})$ in case of strong dependence; $X_3(\boldsymbol{s}) \geq 0$ is a spatial process with unit mean and non-trivial spatial dependence structure, which captures the decay of spatial dependence with respect to spatial distance; $X_1(\boldsymbol{s})$, $X_2(\boldsymbol{s})$ and $X_3(\boldsymbol{s})$ are mutually independent; and $\alpha(\boldsymbol{s}) > 0$ is a spatially-varying scale parameter capturing non-stationarity in terms of covariates. Note that $\alpha(\boldsymbol{s}) = \mathbb{E}[Y(\boldsymbol{s})]$, owing to the unit mean condition on the random product terms; see Section 2.1 for more details. Through their marginal distributions, these three underlying random processes are appropriately weighted to determine the extent to which they contribute to the overall product mixture. We carefully design the roles of the three components to ensure their identifiability and to allow for meaningful interpretations. Our constructive modeling framework provides flexible models for the upper tail, and sub-asymptotic levels, such that we may even use them for modeling complete datasets with heavy-tailed margins and moderate to strong upper-tail dependence. In our models, the strength of

tail dependence relies on the choice of the underlying copula in the latent process $X_3(\boldsymbol{s})$, thus giving flexibility to capture various asymptotic dependence regimes; see Section 2.2 for more details.

In this work, we focus on modeling heavy-tailed data, though the model could also be adapted to light tails with exponential decay by switching to an additive structure via a logarithmic transformation of (2), i.e., the data would be represented as $\log \alpha(\boldsymbol{s}) + \log X_1(\boldsymbol{s}) + \log X_2(\boldsymbol{s}) + \log X_3(\boldsymbol{s})$. Negative values can arise in this construction, but this could be avoided by considering the so-called softplus transformation $\log(1 + Y(\boldsymbol{s}))$ instead of $\log Y(\boldsymbol{s})$, with $Y(\boldsymbol{s})$ as in (2).

To fit our model to threshold exceedances $Y(\boldsymbol{s}) > u(\boldsymbol{s})$, where $u(\boldsymbol{s})$ is some high spatially-varying threshold, we take advantage of the hierarchical formulation of our model, and we exploit customized MCMC methods. Specifically, we use the simulation-based Metropolis adjusted Langevin algorithm (MALA) with random block proposals for latent parameters, as well as the stochastic gradient Langevin dynamics (SGLD) algorithm (Welling and Teh, 2011) for hyperparameters. Our SGLD algorithm combines the popular stochastic gradient descent algorithm, known as an important optimization method in Machine Learning (Neal, 2012; Deng et al., 2018; Zhang et al., 2020), and the Langevin dynamics (Neal, 2011), to tackle problems in relatively high spatio-temporal dimensions. The SGLD algorithm indeed significantly reduces the computation burden to fit our model to threshold exceedances, and allows inference for massive datasets, by contrast to alternative, more classical inference methods for extreme-value POT models (Thibaud and Opitz, 2015; de Fondeville and Davison, 2018; Huser and Wadsworth, 2019). In our modeling framework, low values (such that $Y(\boldsymbol{s}) \leq u(\boldsymbol{s})$) are completely censored in the MCMC algorithm. This censoring mechanism can be performed very efficiently thanks to the independent noise component $X_1(\boldsymbol{s})$ in (2); see also Yadav et al. (2021) and Zhang et al. (2021) for a related censored inference approach.

The paper is organized as follows. In Section 2, we define our general product mixture model for spatial extremes with a specific example and study its joint tail behavior. In Section 3, we provide

censored likelihood expressions for the product mixture model and a simulation study to check the performance of the MCMC sampler based on the MALA/SGLD algorithm, and in Section 4, we fit our model to daily mean precipitation intensities in North-Eastern Spain observed at 94 monitoring stations. Conclusions and some future research directions are detailed in Section 5.

2 Product mixture models for spatial threshold exceedances

2.1 General construction

Let $Y_t(\boldsymbol{s})$, $\boldsymbol{s} \in \mathcal{S} \in \mathbb{R}^2$, be the spatial process of interest observed at times $t \in \{1, \ldots, n\}$ and at a finite set of d locations $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_d \in \mathcal{S}$ with study region $\mathcal{S} \subset \mathbb{R}^2$, and let the random vector $\boldsymbol{Y}_t = (Y_{t1}, \ldots, Y_{td})^T = \{Y_t(\boldsymbol{s}_1), \ldots, Y_t(\boldsymbol{s}_d)\}^T$ denote the t^{th} observed time replicate. We assume that the processes $Y_t(\boldsymbol{s})$, $t = 1, \ldots, n$, are iid copies of the process $Y(\boldsymbol{s})$ in (2), such that $Y_t(\boldsymbol{s}) = \alpha(\boldsymbol{s}) X_{1t}(\boldsymbol{s}) X_{2t}(\boldsymbol{s}) X_{3t}(\boldsymbol{s})$, for some non-negative independent processes $X_{1t}(\boldsymbol{s}), X_{2t}(\boldsymbol{s})$ and $X_{3t}(\boldsymbol{s})$. Similarly, we write $\boldsymbol{X}_{1t} = (X_{1t1}, \ldots, X_{1td})^T = \{X_{1t}(\boldsymbol{s}_1), \ldots, X_{1t}(\boldsymbol{s}_d)\}^T$, $\boldsymbol{X}_{2t} = (X_{2t1}, \ldots, X_{2td})^T = \{X_{2t}(\boldsymbol{s}_1), \ldots, X_{2t}(\boldsymbol{s}_d)\}^T$, $\boldsymbol{X}_{3t} = (X_{3t1}, \ldots, X_{3td})^T = \{X_{3t}(\boldsymbol{s}_1), \ldots, X_{3t}(\boldsymbol{s}_d)\}^T$, and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_d)^T = \{\alpha(\boldsymbol{s}_1), \ldots, \alpha(\boldsymbol{s}_d)\}^T$. Moreover, with some slight abuse of notation, we write interchangeably $Y \sim F$, or $Y \sim F(\cdot)$ when more convenient, to indicate that Y has distribution function F, i.e., $\Pr(Y \leq y) = F(y)$, and similarly for multivariate distributions. We now describe each of the terms in the above product mixture. We assume that $X_{1t}(\boldsymbol{s})$ is a noise process such that the variables $X_{1t}(\boldsymbol{s}_j)$ are iid with distribution function F_1 and mean one, i.e., $\mathbb{E}(X_{1tj}) = 1$, for $j = 1, \ldots, d, t = 1 \ldots, n$. We further assume that $X_{2t}(\boldsymbol{s})$ is spatially constant, such that, almost surely, $X_{2t1} = \cdots = X_{2td} \equiv X_{2t} \overset{\text{iid}}{\sim} F_2$, for some distribution F_2, and that $\mathbb{E}(X_{2t}) = 1$ for $t = 1, \ldots, n$. Specifically, we assume that F_1 and F_2 have Weibull-like tails, i.e., $1 - F_1(x) \sim c x^{\eta} \exp\{-(x/\lambda)^{\kappa}\}$, as $x \to \infty$, for some $\eta \in \mathbb{R}$, $\lambda > 0$, Weibull coefficient $\kappa > 0$ and a constant $c > 0$, and similarly for F_2 with the same structure but potentially different parameters. Finally, we assume that $X_{3t}(\boldsymbol{s})$ is a non-trivial spatial process such that $\boldsymbol{X}_{3t} \overset{\text{iid}}{\sim} F_{\boldsymbol{X}_3}$, where the joint

distribution $F_{\boldsymbol{X}_3}$ has an underlying copula $C_{\boldsymbol{X}_3}$ (i.e., multivariate distribution with fixed uniform margins) and regularly varying marginal distribution, F_3, with index $-1/\xi < 0$, for some $0 < \xi < 1$, and $\mathbb{E}(X_{3tj}) = 1$, $j = 1, \ldots, d$, $t = 1, \ldots, n$. Thus, we have that $1 - F_1(x) = o(1 - F_3(x))$ and $1 - F_2(x) = o(1 - F_3(x))$, as $x \to \infty$, such that the tails of F_1 and F_2 are dominated by the tail of F_3. We assume that the three random fields $X_{1t}(\boldsymbol{s})$, $X_{2t}(\boldsymbol{s})$, and $X_{3t}(\boldsymbol{s})$ are mutually independent (also across different times t). As for the spatially-varying scale parameter $\alpha(\boldsymbol{s})$, we model it with covariates through a log-link function. Therefore, our general product mixture model is defined at the observed sites as

$$\boldsymbol{Y}_t = \boldsymbol{\alpha} \boldsymbol{X}_{1t} \boldsymbol{X}_{2t} \boldsymbol{X}_{3t}, \quad t = 1, \ldots, n, \tag{3}$$

where $\boldsymbol{\alpha} = \exp(\gamma_0 \mathbf{1} + \gamma_1 \mathbf{Z}_1 + \cdots + \gamma_p \mathbf{Z}_p)$ is the scale vector, $\mathbf{Z}_1, \ldots, \mathbf{Z}_d$ are spatial covariates measured at the d locations, $\boldsymbol{\Theta}_{\boldsymbol{\alpha}} = (\gamma_0, \gamma_1, \ldots, \gamma_p)^T$ are the corresponding regression coefficients, n denotes the total number of independent (ind) time replicates, and operations are done componentwise. By construction, the mean of the observed vector $\boldsymbol{Y}_t$ in (3) is $\mathbb{E}(\mathbf{Y}_t) = \boldsymbol{\alpha}$, and the marginal tail index is ξ (thanks to Breiman's Lemma (1)). The three random vectors in the product model (3) are designed to have distinct roles and capture different characteristics. Specifically, the random vector $\boldsymbol{X}_{1t}$ is composed of iid variables, which allows us to capture small-scale variations and to perform fast Bayesian computations in case non-extreme observations are censored. The term $\boldsymbol{X}_{2t}$ has a fully dependent spatial structure, which counterbalances the iid term $\boldsymbol{X}_{1t}$ in case the data have an overall strong spatial dependence. Finally, $\boldsymbol{X}_{3t}$ has a non-trivial spatial dependence structure, which is needed to capture the decay of spatial dependence with respect to distance. By suitably defining their marginal distributions, F_1, F_2, and F_3, respectively, each of these random fields are appropriately weighted in our model with some "weight" parameters to be estimated from the data. Some specific examples will be given in Section 2.3.

From (3), we can rewrite the spatial product mixture model hierarchically as follows:

$$
\begin{aligned}
Y_{tj} \mid \boldsymbol{X}_{2t}, \boldsymbol{X}_{3t}, \boldsymbol{\Theta}_{\boldsymbol{X}_1}, \boldsymbol{\Theta}_{\boldsymbol{\alpha}} &\overset{\text{ind}}{\sim} F_1(\,\cdot\,/(\alpha_j X_{3tj} X_{2t})\,; \boldsymbol{\Theta}_{\boldsymbol{X}_1}), \qquad j = 1, \ldots, d,\ t = 1, \ldots, n; \\
X_{2t} \mid \boldsymbol{\Theta}_{\boldsymbol{X}_2} &\overset{\text{ind}}{\sim} F_2(\,\cdot\,; \boldsymbol{\Theta}_{\boldsymbol{X}_2}), \qquad t = 1, \ldots, n; \\
\boldsymbol{X}_{3t} \mid \boldsymbol{\Theta}_{\boldsymbol{X}_3}^{\text{mar}^T}, \boldsymbol{\Theta}_{\boldsymbol{X}_3}^{\text{dep}^T} &\overset{\text{ind}}{\sim} C_{\boldsymbol{X}_3}\left\{F_3(\,\cdot\,; \boldsymbol{\Theta}_{\boldsymbol{X}_3}^{\text{mar}}), \ldots, F_3(\,\cdot\,; \boldsymbol{\Theta}_{\boldsymbol{X}_3}^{\text{mar}}); \boldsymbol{\Theta}_{\boldsymbol{X}_3}^{\text{dep}}\right\}; \\
\boldsymbol{\Theta} &\sim \pi(\boldsymbol{\Theta}),
\end{aligned}
\tag{4}
$$

where $\boldsymbol{\Theta} = (\boldsymbol{\Theta}_{\boldsymbol{\alpha}}^T, \boldsymbol{\Theta}_{\boldsymbol{X}_1}^T, \boldsymbol{\Theta}_{\boldsymbol{X}_2}^T, \boldsymbol{\Theta}_{\boldsymbol{X}_3}^{\text{mar}^T}, \boldsymbol{\Theta}_{\boldsymbol{X}_3}^{\text{dep}^T})^T$ are the unknown model hyperparameters, $\boldsymbol{\Theta}_{\boldsymbol{X}_1}$ and $\boldsymbol{\Theta}_{\boldsymbol{X}_2}$ denote the hyperparameter vectors for the marginal distributions of the random vectors $\boldsymbol{X}_{1t}$ and $\boldsymbol{X}_{2t}$, respectively, $\boldsymbol{\Theta}_{\boldsymbol{X}_3} = (\boldsymbol{\Theta}_{\boldsymbol{X}_3}^{\text{mar}^T}, \boldsymbol{\Theta}_{\boldsymbol{X}_3}^{\text{dep}^T})^T$ contains parameters for the random vector $\boldsymbol{X}_{3t}$, with $\boldsymbol{\Theta}_{\boldsymbol{X}_3}^{\text{mar}^T}$ controlling its marginal distribution and $\boldsymbol{\Theta}_{\boldsymbol{X}_3}^{\text{dep}^T}$ controlling the dependence structure, $\pi(\boldsymbol{\Theta})$ denotes the prior distribution for the hyperparameter vector $\boldsymbol{\Theta}$, and α_j denotes the j^{th} parameter in the scale vector $\boldsymbol{\alpha}$, which depends on $\boldsymbol{\Theta}_{\boldsymbol{\alpha}}^T = (\gamma_0, \gamma_1, \ldots, \gamma_p)^T$.

The hierarchical construction of the proposed product model (4) suggests using Bayesian inference based on the full data vector $\boldsymbol{Y} = (\boldsymbol{Y}_1^T, \ldots, \boldsymbol{Y}_n^T)^T$, by treating $\boldsymbol{X}_2 = (X_{21}, \ldots X_{2n})^T$ and $\boldsymbol{X}_3 = (\boldsymbol{X}_{31}^T, \ldots, \boldsymbol{X}_{3n}^T)^T$ as latent variables. These two latent vectors, $\boldsymbol{X}_2$ and $\boldsymbol{X}_3$, are of dimension n and nd, respectively. Therefore, there will be a total of $n + nd + |\boldsymbol{\Theta}|$ latent variables and hyperparameters to make inference for, simultaneously. Let $\pi(\cdot)$ denote a generic (conditional) density, then the joint posterior density of $\boldsymbol{\Theta}$, $\boldsymbol{X}_2$, and $\boldsymbol{X}_3$, denoted by $\pi_{\text{post}}(\boldsymbol{\Theta}, \boldsymbol{X}_2, \boldsymbol{X}_3 \mid \boldsymbol{Y})$, is proportional to $\pi(\boldsymbol{Y}, \boldsymbol{\Theta}, \boldsymbol{X}_2, \boldsymbol{X}_3) = \pi(\boldsymbol{Y} \mid \boldsymbol{X}_2, \boldsymbol{X}_3, \boldsymbol{\Theta}_{\boldsymbol{X}_1}, \boldsymbol{\Theta}_{\boldsymbol{\alpha}})\pi(\boldsymbol{X}_2 \mid \boldsymbol{\Theta}_{\boldsymbol{X}_2})\pi(\boldsymbol{X}_3 \mid \boldsymbol{\Theta}_{\boldsymbol{X}_3})\pi(\boldsymbol{\Theta})$, and the posterior density of $\boldsymbol{\Theta}$ is thus obtained after integrating out the latent variables $\boldsymbol{X}_2$ and $\boldsymbol{X}_3$, i.e.,

$$
\pi(\boldsymbol{\Theta} \mid \boldsymbol{Y}) = \iint \pi_{\text{post}}(\boldsymbol{\Theta}, \boldsymbol{X}_2, \boldsymbol{X}_3 \mid \boldsymbol{Y}) \mathrm{d}\boldsymbol{X}_2 \mathrm{d}\boldsymbol{X}_3. \tag{5}
$$

The dimension of the integral in (5) is very large. We solve this issue in Section 3.3 by using a customized MCMC algorithm that combines the Metropolis adjusted Langevin algorithm (MALA) with block proposals and the stochastic gradient Langevin dynamics (SGLD), in order to efficiently

generate representative posterior samples of $\boldsymbol{\Theta}$, $\boldsymbol{X}_2$, and $\boldsymbol{X}_3$ from the target posterior distribution.

2.2 Joint tail behavior

We here derive the theoretical joint tail behavior of the spatial product mixture model (3) for the case where the latent vector $\boldsymbol{X}_{3t}$ has a regularly varying marginal distribution with positive tail index $\xi > 0$ and $\boldsymbol{X}_{1t}$ and $\boldsymbol{X}_{2t}$ are lighter-tailed such that $\mathbb{E}\{(X_{1tj}X_{2tj})^{1/\xi+\varepsilon}\} = \mathbb{E}(X_{1tj}^{1/\xi+\varepsilon})\mathbb{E}(X_{2tj}^{1/\xi+\varepsilon}) < \infty$ for some $\varepsilon > 0$, which includes the main specific example detailed in Section 2.3. For convenience, we drop the subscript t in this subsection, so that the spatial product mixture model (3) may be generally written as $\boldsymbol{Y} = \boldsymbol{\alpha}\boldsymbol{X}_1\boldsymbol{X}_2\boldsymbol{X}_3$, with $\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3$ defined as above. Furthermore, let the multivariate distribution $F_{\boldsymbol{X}_3}$ of $\boldsymbol{X}_3$ at a finite number of locations be (jointly) regularly varying at ∞ (Resnick, 1987) such that

$$\frac{1 - F_{\boldsymbol{X}_3}(z\boldsymbol{x}_3)}{1 - F_{\boldsymbol{X}_3}(z\boldsymbol{1})} \to V_{\boldsymbol{X}_3}(\boldsymbol{x}_3), \quad \boldsymbol{x}_3 > \boldsymbol{0}, \quad z \to \infty,$$

where $\boldsymbol{1} = (1, \ldots, 1)^T \in \mathbb{R}^d$ and $V_{\boldsymbol{X}_3}(\cdot)$ is some positive limit function that is homogeneous of order $-1/\xi$, i.e., $V_{\boldsymbol{X}_3}(z\boldsymbol{x}_3) = z^{-1/\xi}V_{\boldsymbol{X}_3}(\boldsymbol{x}_3)$ for all $\boldsymbol{x}_3 > \boldsymbol{0}$ and $z > 0$. Then, Theorem 3 of Fougeres and Mercadier (2012) implies the multivariate regular variation of $F_{\boldsymbol{Y}}$, which denotes the multivariate distribution of the data vector $\boldsymbol{Y}$, i.e.,

$$\frac{1 - F_{\boldsymbol{Y}}(z\boldsymbol{y})}{1 - F_{\boldsymbol{Y}}(z\boldsymbol{1})} \to V_{\boldsymbol{Y}}(\boldsymbol{y}) = \int_0^\infty \int_0^\infty \cdots \int_0^\infty V_{\boldsymbol{X}_3}\{\boldsymbol{y}/(\boldsymbol{\alpha}x_2\boldsymbol{x}_1)\} \left(\prod_{j=1}^{d} \mathrm{d}F_1(x_{1j}) \right) \mathrm{d}F_2(x_2), \quad z \to \infty, \tag{6}$$

where $V_{\boldsymbol{Y}}(\cdot)$ is some positive limit function that is also homogeneous of order $-1/\xi$. Equation (6) fully characterizes the extremal dependence of the product mixture $\boldsymbol{Y}$ resulting from (3) in the heavy-tailed case. More, explicitly, the bivariate random vector $\boldsymbol{Y} = (Y_1, Y_2)^T$ is asymptotically independent if and only if $\boldsymbol{X}_3 = (X_{31}, X_{32})^T$ is asymptotically independent; note that asymptotic independence in the bivariate vector $\boldsymbol{Y}$ is equivalent to $V_{\boldsymbol{Y}}(y_1, y_2) = c_V\{(y_1/\alpha_1)^{-1/\xi} + (y_2/\alpha_2)^{-1/\xi}\}$ with a constant $c_V > 0$ and without any sum terms where y_1 and y_2 appear together. Therefore, the asymptotic dependence class of the product mixture $\boldsymbol{Y}$ depends on the choice of copula in the

latent vector $\boldsymbol{X}_3$. For example, $\boldsymbol{Y}$ is asymptotically independent when we use a Gaussian copula in $\boldsymbol{X}_3$ and asymptotically dependent when we use a Student's t copula with $\nu > 0$ degrees of freedom instead. Let $Y_1 \sim F_{Y_1}$ and $Y_2 \sim F_{Y_2}$, then a summary of the extremal dependence strength is the tail correlation coefficient $\chi = \lim_{u \to 1} \chi(u)$, where $\chi(u) = \Pr\{Y_1 > F_{Y_1}^{-1}(u) \mid Y_2 > F_{Y_2}^{-1}(u)\}$; this coefficient is linked to the function $V_{\boldsymbol{Y}}(\cdot)$ via $\chi = 1 - V_{\boldsymbol{Y}}(\alpha_1, \alpha_2)/V_{\boldsymbol{Y}}(\alpha_1, \infty)$, and will be illustrated graphically for one of the specific models proposed in Section 2.3.

2.3 Examples of flexible product mixture models

The general product mixture model formulation in (3) can be used to construct various specific spatial models with a flexible heavy-tailed behavior in the upper tail. Here, we detail one specific example of spatial product mixture model, which we also use in our simulation study in Section 3, and in the data application in Section 4, and then discuss further alternative possibilities, as well. Let Exp(1) denote the exponential distribution with rate parameter one. Then, a particular product mixture model can be obtained by specifying the terms in (3) as follows:

$$X_{1tj} = E_{1tj}^{\beta_1}/\Gamma(1+\beta_1), \quad E_{1tj} \overset{iid}{\sim} \text{Exp}(1), \quad j = 1 \ldots, d,\, t = 1, \ldots, n, \quad \beta_1 > 0;$$
$$X_{2t} = E_{2t}^{\beta_2}/\Gamma(1+\beta_2), \quad E_{2t} \overset{iid}{\sim} \text{Exp}(1), \quad t = 1, \ldots, n, \quad \beta_2 > 0,$$

where $\Gamma(x) = \int_0^\infty s^{x-1}e^{-s}\text{ds}$ is the gamma function for $x > 0$. The parameters β_1 and β_2 affect the marginal distributions of $\boldsymbol{X}_{1t}$ and $\boldsymbol{X}_{2t}$, respectively, and can be interpreted as "weights" in the mixture (3), though they need not sum to one nor be less than one. As $\beta_1 \to 0$, the independent term $\boldsymbol{X}_{1t}$ indeed becomes degenerate at one, and is thus negligible in the mixture (3). This holds similarly for the fully dependent term $\boldsymbol{X}_{2t}$, as $\beta_2 \to 0$. To ensure tails that are not heavier than exponential and/or to prevent potential numerical issues when computing the gradient of the log-posterior distribution (see Section 3 and the Supplementary Material), both β_1 and β_2 may be restricted to the interval $(0, 1]$, though this is not strictly necessary and in the application in Section 4.1 we allow slightly heavier-tailed distributions by restricting β_1 and β_2 to $(0, 2]$ instead.

As for the random vector $\boldsymbol{X}_{3t}$, we define its marginal distribution F_3 to be the Inverse Gamma (IG) distribution with scale $\beta_3 - 1$, and shape β_3, for some parameter $\beta_3 > 1$, such that $\mathbb{E}(X_{3tj}) = 1$, as desired. Moreover, as $\beta_3 \to \infty$, the term $\boldsymbol{X}_{3t}$ becomes degenerate at one and does not have any effect on the mixture (3). Thus, this spatial model is suitable for moderately heavy-tailed data with $\xi = 1/\beta_3 > 0$ (i.e., $\beta_3 < \infty$). In summary, the marginal distributions F_1, F_2, and F_3 in (4) can be written as

$$F_1(\cdot) = \text{Wb}\left\{\cdot\,; \frac{1}{\beta_1}, \frac{1}{\Gamma(1+\beta_1)}\right\}, F_2(\cdot) = \text{Wb}\left\{\cdot\,; \frac{1}{\beta_2}, \frac{1}{\Gamma(1+\beta_2)}\right\}, F_3(\cdot) = \text{IG}(\cdot\,; \beta_3, \beta_3 - 1), \quad (7)$$

where $\text{Wb}(\cdot\,; \kappa, \lambda)$ denotes the Weibull distribution function with shape $\kappa > 0$ and scale $\lambda > 0$, and $\text{IG}(\cdot\,; a, b)$ denotes the Inverse Gamma distribution function with shape $a > 0$ and scale $b > 0$. Furthermore, let the underlying copula $C_{\boldsymbol{X}_3}$ of $\boldsymbol{X}_{3t}$ be the Gaussian copula with exponential correlation function $\rho(h) = \exp(-h/\rho)$, $h \geq 0$, and range $\rho > 0$. Then, $\boldsymbol{X}_{3t}$ has joint distribution

$$\boldsymbol{X}_{3t} \mid \boldsymbol{\Theta}_{\boldsymbol{X_3}}^{\text{mar}^T}, \boldsymbol{\Theta}_{\boldsymbol{X_3}}^{\text{dep}^T} \sim \Phi_\rho\Big(\Phi^{-1}\big[\text{IG}\{\cdot\,; \beta_3, \beta_3 - 1\}\big], \ldots, \Phi^{-1}\big[\text{IG}\{\cdot\,; \beta_3, \beta_3 - 1\}\big]\Big), \quad (8)$$

where Φ_ρ is the multivariate Gaussian distribution function with zero mean and correlation matrix $\boldsymbol{\Sigma}(\rho)$ with entries $\boldsymbol{\Sigma}_{i_1,i_2} = \exp(-\|\boldsymbol{s}_{i_1} - \boldsymbol{s}_{i_2}\|/\rho)$, and Φ^{-1} is the quantile function of the standard Gaussian distribution. For this specific model, the hyperparameter vector is $\boldsymbol{\Theta} = (\boldsymbol{\Theta}_{\boldsymbol{\alpha}}^T, \boldsymbol{\Theta}_{\boldsymbol{X}_1}^T, \boldsymbol{\Theta}_{\boldsymbol{X}_2}^T, \boldsymbol{\Theta}_{\boldsymbol{X}_3}^{\text{mar}^T}, \boldsymbol{\Theta}_{\boldsymbol{X}_3}^{\text{dep}^T})^T$ with

$$\boldsymbol{\Theta}_{\boldsymbol{\alpha}} = (\gamma_0, \gamma_1, \ldots, \gamma_p)^T, \quad \boldsymbol{\Theta}_{\boldsymbol{X}_1} = \beta_1, \quad \boldsymbol{\Theta}_{\boldsymbol{X}_2} = \beta_2, \quad \boldsymbol{\Theta}_{\boldsymbol{X}_3}^{\text{mar}} = \beta_3, \quad \text{and} \quad \boldsymbol{\Theta}_{\boldsymbol{X}_3}^{\text{dep}} = \rho.$$

The marginal tail index for this model is $\xi = 1/\beta_3 > 0$, as the $\text{IG}(\cdot\,; a, b)$ distribution is regularly varying with index a. Therefore, as described, the marginal distribution of $\boldsymbol{Y}$ is heavy-tailed, and the tail heaviness is controlled through the parameter β_3.

One limitation behind the specific model proposed above is that the asymptotic tail dependence class is fixed a priori, and determined by the underlying Gaussian copula. Specifically, the construc-

tion above yields asymptotic independence. As mentioned in Section 2.2, we can get asymptotic dependence by choosing a latent Student's t copula instead, but the tail dependence class would still be predetermined, and not inferred from the data. Although we may extend our modeling framework by allowing more complex types of copulas that bridge tail dependence and independence classes (e.g., Huser et al., 2017; Huser and Wadsworth, 2019), implementation and computations would become more involved. Therefore, in this work, we only consider latent Gaussian or Student's t copulas, which already provide a rich and computationally convenient class of quite flexible dependence structures. Nevertheless, note that when $\boldsymbol{X}_3$ is characterized by an (asymptotically dependent) Student's t copula with dispersion matrix $\boldsymbol{\Sigma}(\rho)$ and degrees of freedom $\nu > 0$, the dependence strength decreases as ν increases and eventually reduces to the (asymptotically independent) Gaussian copula as $\nu \to \infty$. Therefore, a relatively simple way to bridge tail dependence classes within our framework could be to let $\nu' = 1/\nu$ and assign an informative prior to ν' that pushes the Student's t copula model towards $\nu' = 0$ ($\nu = \infty$), i.e., asymptotic independence. Such a prior could even assign positive prior probability to $\nu' = 0$. By doing this, our model can yield positive posterior probability to both asymptotic dependence types, thus providing a data-driven way to infer the limiting tail structure. We leave this interesting direction to future research.

We now study the extremal dependence strength of the product mixture model (7) in more detail, in terms of the tail correlation coefficients $\chi(u)$ and χ when using a latent Gaussian copula, or a latent Student's t copula with $\nu > 0$ degrees of freedom. Figure 1 shows the plot of $\chi(u)$ as a function of u and its limit χ (first column), and the marginal histogram (last two-columns) when using a latent Gaussian copula (first row) and latent Student's t copula (second row), with an underlying exponential correlation function with range $\rho = 1$, at a spatial distance 0.5 (i.e., for a correlation of $\exp(-0.5) \approx 0.61$ in the latent vector $\boldsymbol{X}_3$). The other hyperparameters are set as follows: $\alpha_1 = \alpha_2 = 1$, $(\beta_1, \beta_2)^T = (0.25, 0.75)^T$ or $(0.75, 0.25)^T$, and for the Student's t case, $\nu = 1$. These plots demonstrate that our product model can indeed capture various tail dependence

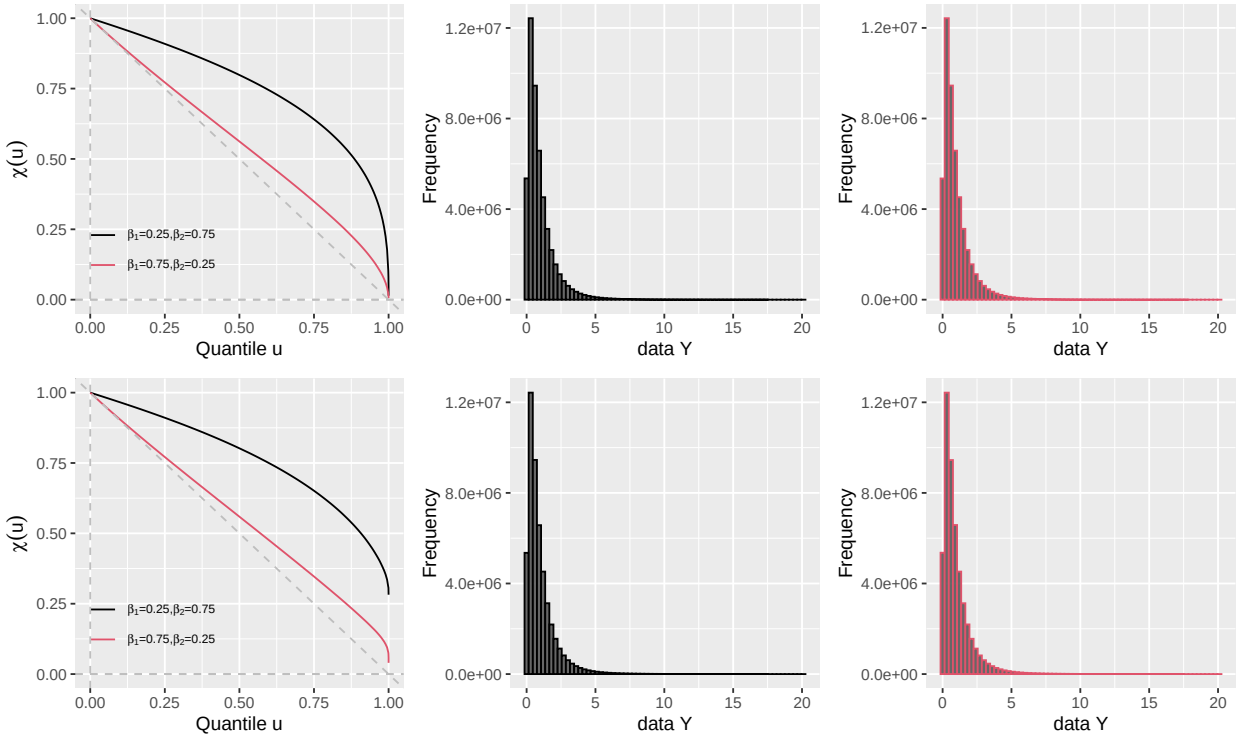

Figure 1: Plot of $\chi(u)$ (first column) for the product mixture model (3) with specific structure detailed in Section 2.3, based on a latent Gaussian copula (top) or Student's t copula (bottom), with parameters chosen as $(\beta_1, \beta_2) = (0.25, 0.75)$ (black) or $(0.75, 0.25)$ (red), $\alpha_1 = \alpha_2 = 1$, $\beta_3 = 5$, $\rho = 1$, and for Student's t case, $\nu = 1$, as well as the respective marginal histograms (second and third columns).

decays, with stronger tail dependence when β_2 is higher. This is expected as β_2 is the "weight" associated with the fully dependent term $\boldsymbol{X}_2$ in the mixture model (7). Also, all the histograms appear to be quite similar to each other, which shows that the different parameter combinations give flexibility to capture different types of joint tail behavior, while the marginal tail behavior remains relatively unaffected when the scale parameters α_1, α_2, and the shape parameter β_3 (i.e., the reciprocal of the tail index) are kept fixed. Moreover, as expected, the dependence strength of the data vector $\boldsymbol{Y}$ depends on the choice of copula in the latent vector $\boldsymbol{X}_3$, i.e., the limiting tail correlation coefficient, χ, is strictly positive when we use a latent Student's t copula, whereas $\chi = 0$ when we use a latent Gaussian copula instead.

The specific distributions proposed in (7) can be modified in different ways to create alternative

product mixture models of the form (3). One interesting possibility is to keep F_1 and F_2 as defined in (7) but replace F_3 with a (unit mean) generalized Pareto (GP) distribution with scale $1-\xi$ and shape ξ for some $\xi < 1$. When $\xi > 0$, F_3 is regularly varying with index $-1/\xi$, hence the marginal distribution of the product process (3) is heavy-tailed with tail index ξ. When $\xi \to 0$, F_3 has an exponential tail behavior and hence the marginal tail behavior of the product process is of Weibull type instead, with Weibull index determined by the interplay between β_1 and β_2. When $\xi < 0$, F_3 has a bounded upper tail, so the process X_3 is dominated by the other two processes X_1 and X_2 and becomes completely irrelevant in the limiting tail. An extension of this model, which is more flexible to capture Weibull-like tails, is to take F_3 as the Burr distribution (rescaled to have mean one), which essentially corresponds to the distribution of positive powers of GP variables.

3 Simulation-based Bayesian inference

3.1 General strategy

The hierarchical construction (4) of our proposed spatial product mixture model (3) naturally suggests using simulation-based Bayesian inference where latent variables are simulated conditional on observations. We use Markov chain Monte Carlo (MCMC) methods to simultaneously generate samples of the hyperparameter vector $\boldsymbol{\Theta}$ and the two latent parameter vectors, $\boldsymbol{X}_2$ and $\boldsymbol{X}_3$. As we fit our model to threshold exceedances, we describe in Section 3.2 the censoring mechanism focusing on the specific model described in Section 2.3. In Section 3.3, we detail our MCMC sampler combining block Metropolis adjusted Langevin algorithm (MALA) updates for latent variables and the stochastic gradient Langevin dynamics (SGLD) for hyperparameters. We demonstrate the performance of our MCMC sampler based on a simulation study in Section 3.4.

3.2 Censored likelihood with latent variables, priors, and posterior density

We follow the notation of Section 2.1, with lowercase letters denoting realized values, i.e., y_{tj} is the realization of $Y_{tj} = Y_t(\boldsymbol{s}_j)$, x_{1tj} is the realization of $X_{1tj} = X_{1t}(\boldsymbol{s}_j)$, and so forth. Let

$\boldsymbol{e}_t = (e_{t1}, \ldots, e_{td})^T$ be the exceedance indicator vector, such that $e_{tj} = 1$, if $y_{tj} > u_{tj}$, and $e_{tj} = 0$, if $y_{tj} \leq u_{tj}$, where $\boldsymbol{u}_t = (u_{t1}, \ldots, u_{td})^T$ is a fixed threshold vector. If $u_{tj} = 0$, no censoring is applied to the value y_{tj}, whereas if $u_{tj} = \infty$, the observation y_{tj} is treated as fully censored and as a variable to predict. Then, the augmented censored likelihood contribution for the parameter $(\boldsymbol{\Theta}^T, x_{2t}, \boldsymbol{x}_{3t}^T)^T$, based on the data vector $(\boldsymbol{y}_t^T, \boldsymbol{e}_t^T)^T$, which stems from the general product mixture model (3) with an underlying copula $C_{\boldsymbol{X}_3}$ (with density $c_{\boldsymbol{X}_3}$), is

$$
\begin{aligned}
L(\boldsymbol{\Theta}, x_{2t}, \boldsymbol{x}_{3t}; \boldsymbol{y}_t, \boldsymbol{e}_t) = & \prod_{j=1}^{d} \left\{ \frac{1}{\alpha_j x_{2t} x_{3t}} f_1 \left(\frac{y_{tj}}{\alpha_j x_{2t} x_{3tj}}; \boldsymbol{\Theta}_{\boldsymbol{X}_1} \right) \right\}^{\mathbb{I}(e_{tj}=1)} \\
& \times \prod_{j=1}^{d} \left\{ F_1 \left(\frac{u_{tj}}{\alpha_j x_{2t} x_{3tj}}; \boldsymbol{\Theta}_{\boldsymbol{X}_1} \right) \right\}^{\mathbb{I}(e_{tj}=0)} \\
& \times f_2(x_{2t}; \boldsymbol{\Theta}_{\boldsymbol{X}_2}) \\
& \times c_{\boldsymbol{X}_3} \left\{ F_3(x_{3t1}; \boldsymbol{\Theta}_{\boldsymbol{X}_3}^{\text{mar}}), \ldots, F_3(x_{3td}; \boldsymbol{\Theta}_{\boldsymbol{X}_3}^{\text{mar}}); \boldsymbol{\Theta}_{\boldsymbol{X}_3}^{\text{dep}} \right\} \prod_{j=1}^{d} f_3(x_{3tj}; \boldsymbol{\Theta}_{\boldsymbol{X}_3}^{\text{mar}}), \qquad (9)
\end{aligned}
$$

where f_1, f_2 and f_3 are the density functions corresponding to the cumulative distribution function F_1, F_2 and F_3, respectively, and $\mathbb{I}(\cdot)$ is the indicator function. The expressions in (9) correspond to the likelihood contribution of non-censored observations (first line) and censored observations (second line), to the likelihood contribution of the latent variable x_{2t} (third line) and the latent variable vector $\boldsymbol{x}_{3t}$ (fourth line). In particular, when $C_{\boldsymbol{X}_3}$ is the Gaussian copula, the density $c_{\boldsymbol{X}_3}$ has the form

$$
c_{\boldsymbol{X}_3}(\boldsymbol{v}) = \phi_\rho \left\{ \Phi^{-1}(v_1), \ldots, \Phi^{-1}(v_d) \right\} \left[\prod_{j=1}^{d} \phi\{\Phi^{-1}(v_j)\} \right]^{-1}, \quad \boldsymbol{v} = (v_1, \ldots, v_d)^T \in [0,1]^d,
$$

where ϕ_ρ and ϕ are the multivariate and univariate Gaussian densities corresponding to Φ_ρ and Φ introduced in (8). Assuming independent time replicates, the overall augmented censored likelihood is then

$$
L(\boldsymbol{\Theta}, \boldsymbol{x}_2, \boldsymbol{x}_3; \boldsymbol{y}, \boldsymbol{e}) = \prod_{t=1}^{n} L(\boldsymbol{\Theta}, x_{2t}, \boldsymbol{x}_{3t}; \boldsymbol{y}_t, \boldsymbol{e}_t), \qquad (10)
$$

where $\boldsymbol{y} = (\boldsymbol{y}_1^T, \ldots, \boldsymbol{y}_n^T)^T$ is the complete data vector, $\boldsymbol{x}_2 = (x_{21}, \ldots, x_{2n})^T$, $\boldsymbol{x}_3 = (\boldsymbol{x}_{31}^T, \ldots, \boldsymbol{x}_{3n}^T)^T$, and $\boldsymbol{e} = (\boldsymbol{e}_1^T, \ldots, \boldsymbol{e}_n^T)^T$ is the complete threshold indicator vector. In particular, the overall augmented likelihood for the specific model in Section 2.3 may be obtained by using the distribution functions F_1, F_2, F_3, and their corresponding density functions as specified in (7). Using (10), the joint posterior of model parameters $(\boldsymbol{\Theta}^T, \boldsymbol{x}_2^T, \boldsymbol{x}_3^T)^T$ may be written as

$$\pi_{\text{post}}(\boldsymbol{\Theta}, \boldsymbol{x}_2, \boldsymbol{x}_3 \mid \boldsymbol{y}, \boldsymbol{e}) \propto L(\boldsymbol{\Theta}, \boldsymbol{x}_2, \boldsymbol{x}_3; \boldsymbol{y}, \boldsymbol{e})\pi(\boldsymbol{\Theta}). \tag{11}$$

For convenience, we assume independent vague prior distributions for the model hyperparameters. In particular, for the specific model in Section 2.3, we have $\pi(\boldsymbol{\Theta}) = \pi(\gamma_0) \times \pi(\gamma_1) \times \cdots \times \pi(\gamma_p) \times \pi(\beta_1) \times \pi(\beta_2) \times \pi(\beta_3) \times \pi(\rho)$. In the application in Section 4.1, for instance, we choose a standard Gaussian distribution with mean 0 and variance 100 for all the covariate coefficients $\gamma_0, \gamma_1, \ldots, \gamma_p$ controlling the scale vector $\boldsymbol{\alpha} = \exp(\gamma_0\mathbf{1} + \gamma_1\mathbf{Z}_1 + \cdots + \gamma_p\mathbf{Z}_p)$, a uniform distribution on $[0, 1]$ (for simulation study in Section 3.4) or $[0, 2]$ (for data application in Section 4.2) for the "weight" parameters β_1 and β_2 to prevent overly large values, a gamma distribution with shape $1/3$ and rate $1/100$ for β_3 (reciprocal tail index), and a uniform distribution on $[0, 2\delta]$ for ρ (range parameter), where δ is the maximum spatial distance between the stations.

3.3 Stochastic gradient Langevin dynamics (SGLD) algorithm

When writing the model hierarchically, there are two latent parameter vectors, $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$, of dimensions n and nd, respectively. MCMC computations are therefore quite involved, and MCMC chains might mix poorly when n and d are both large. To avoid this issue when the latent parameter vector is of very high dimension, a suitable choice of proposal distributions is required. The Langevin dynamics (Roberts and Tweedie, 1996; Roberts and Rosenthal, 1998) provides an efficient way to define proposal distributions, by exploiting information from the gradient of the log-posterior density, and it generally works quite efficiently in reasonably large dimensions. Let $\pi(\boldsymbol{z} \mid \boldsymbol{y}_n)$ be an arbitrary target posterior density of m components $\boldsymbol{z} = (z_1, \ldots, z_m)^T$ conditioned

on n replicated observations $\boldsymbol{y}_n = (y_1, \ldots, y_n)^T$. Then, the proposal distribution based on the Langevin dynamics requires to fix a step size parameter $\tau > 0$, and to sample proposals $\boldsymbol{z}^p$ from

$$\boldsymbol{z}^p \mid \boldsymbol{z} \sim \mathcal{N}\left(\boldsymbol{z} + \frac{\tau}{2} P \nabla_{\boldsymbol{z}} \log \pi(\boldsymbol{z} \mid \boldsymbol{y}_n),\ \tau P\right), \tag{12}$$

where $\nabla_{\boldsymbol{z}}$ is the gradient operator with respect to the variable $\boldsymbol{z}$ and τP is a covariance matrix; see Atchadé (2006). After a Metropolis–Hastings correction, this algorithm is commonly called the Metropolis adjusted Langevin algorithm, MALA in short. If the dimension m is high, MALA proposals are feasible only if the gradient of the log-posterior density with respect to $\boldsymbol{z}$ is available in closed form and can be computed efficiently. In our case, we have closed-form expressions of the log-posterior density with respect to both latent variable vectors $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$, and some hyperparameters, $\gamma_0, \gamma_1, \ldots, \gamma_p$, β_1 and β_2. For detailed calculations, see Sections 1.2 and 1.3 of the Supplementary Material. One disadvantage of classical MALA proposals, however, is that we need to compute the gradient of the log-posterior density at each MCMC iteration, which may be computationally prohibitive when the data dimension is large. To speed up computations, we instead rely on the SGLD algorithm. The latter may be significantly faster than the MALA, which uses the whole dataset at once (as, e.g., in Yadav et al., 2021). The stochastic gradient descent algorithm was popularized in Machine Learning for fitting complex Neural Network structures. Similarly, it is possible to exploit simulation-based MCMC inference based on the SGLD algorithm (a combination of stochastic gradient descent algorithm, and the Langevin dynamics), in order to speed up computations significantly. This method requires to select a batch size $b \in \{1, \ldots, n\}$ and perform the updates based on a randomly selected sub-dataset of size b instead of the full dataset; for more details see Nemeth and Fearnhead (2020), and the pseudo-code in Algorithm 1. Intuitively, if $b \ll n$, then computations at each iteration will be much faster. More specifically, let $P = I_m$, then the proposal distribution (12) based on sub-dataset of size b may be written as

$$\boldsymbol{z}^p \mid \boldsymbol{z} \sim \mathcal{N}\left(\boldsymbol{z} + \frac{\tau n}{2b} \nabla^{\star}_{\boldsymbol{z}} \log \pi(\boldsymbol{z} \mid \boldsymbol{y}_b),\ \tau I_m\right), \tag{13}$$

Algorithm 1 Pseudo-code for the SGLD algorithm with Metropolis–Hastings correction for the product model of the form (3), written in hierarchical form in (4) and made specific in Section 2.3

Notation: $P \setminus L$: set difference; $|A|$ cardinality of the set A; p: proposed state; c : current state

Input:

- $\boldsymbol{y}_{[A_n,A_d]}$: data matrix of spatial dimension $|A_d|$ and temporal dimension $|A_n|$ where $A_d = \{1,\ldots,d\}$ and $A_n = \{1,\ldots,n\}$.
- $\boldsymbol{\Theta}_k^0$, $\boldsymbol{x}_{2_{[A_n]}}^0$, $\boldsymbol{x}_{3_{[A_n,A_d]}}^0$: initial values for $\boldsymbol{\Theta}_k$, $\boldsymbol{x}_{2_{[A_n]}}$ and $\boldsymbol{x}_{3_{[A_n,A_d]}}$, respectively
- b: batch size for the SGLD algorithm
- N_{burn}: total burn-in samples (for simplicity, we here consider a single burn-in period)
- adapt: number of iterations after which we adapt the tuning parameters
- S_c=$\{(c+0)\times$adapt, $(c+6)\times$adapt, $(c+12)\times$adapt, $\ldots\}$, $c = 1,2,3,4,5,6$
- N_{MH}: number of MCMC iterations after which we apply a Metropolis–Hastings correction
- $N_{tot} = N \times N_{\text{MH}}$: total number of MCMC iterations

Output: Markov chains of hyperparameter vector $\boldsymbol{\Theta} = (\boldsymbol{\Theta}_1^T, \boldsymbol{\Theta}_2^T, \boldsymbol{\Theta}_3^T, \boldsymbol{\Theta}_4^T)^T$, where $\boldsymbol{\Theta}_1 = (\gamma_0, \gamma_1, \ldots, \gamma_p)^T$, $\boldsymbol{\Theta}_2 = \beta_1$, $\boldsymbol{\Theta}_3 = \beta_2$, and $\boldsymbol{\Theta}_4 = (\beta_3, \rho)^T$

Start with $\boldsymbol{\Theta}_k^{c'} = \boldsymbol{\Theta}_k^0$, $\boldsymbol{\Theta}_k^c = \boldsymbol{\Theta}_k^0$, $\boldsymbol{x}_{2_{[A_n]}}^c = \boldsymbol{x}_{2_{[A_n]}}^0$, and $\boldsymbol{x}_{3_{[A_n,A_d]}}^c = \boldsymbol{x}_{3_{[A_n,A_d]}}^0$

for $i = 1$ to N **do**

  for $j = 1$ to N_{MH} **do**

    Randomly sample b indices from A_n without replacement, giving $A_b = \{a_1,\ldots,a_b\} \subset A_n$

    for $k = 1$ to 4 **do**

      SGLD for $\boldsymbol{\Theta}_k$: Propose (transformed) hyperparameter vector $\boldsymbol{\Theta}_k^p$ using SGLD (13) with batch size b, i.e., based on observations $\boldsymbol{y}_{[A_b,A_d]}$, and set $\boldsymbol{\Theta}_k^c = \boldsymbol{\Theta}_k^p$ with probability one

    end for

    Block MALA for $\boldsymbol{x}_2$*:* Propose (log-transformed) $\boldsymbol{x}_{2_{[A_b]}}^p$ using SGLD (13) and calculate the acceptance ratio $r_{\boldsymbol{x}_2} = R_{\boldsymbol{x}_2}(\boldsymbol{x}_{2_{[A_b]}}^p, \boldsymbol{x}_{2_{[A_b]}}^c \mid \boldsymbol{y}_{[A_b,A_d]}, \boldsymbol{e}_{[A_b,A_d]}, \boldsymbol{\Theta}^{c'}, \boldsymbol{x}_{3_{[A_b,A_d]}}^c)$ using (15)

    Generate $U \sim U[0,1]$, **if** $(U < r_{\boldsymbol{x}_2})$ $\{\boldsymbol{x}_{2_{[A_n]}}^c = \boldsymbol{x}_{2_{[A_n \setminus A_b]}}^c \cup \boldsymbol{x}_{2_{[A_b]}}^p\}$ **else** $\{\boldsymbol{x}_{2_{[A_n]}}^c = \boldsymbol{x}_{2_{[A_n]}}^c\}$

    if $((i \times N_{\text{MH}} < N_{\text{burn}})$ & $(i \times N_{\text{MH}} \in S_5))$ {use adaptive strategy to change the step size parameter} **else** {no change}

    Block MALA for $\boldsymbol{x}_3$*:* Propose (log-transformed) $\boldsymbol{x}_{3_{[A_b,A_d]}}^p$ using SGLD (13) and calculate the acceptance ratio $r_{\boldsymbol{x}_3} = R_{\boldsymbol{x}_3}(\boldsymbol{x}_{3_{[A_b,A_d]}}^p, \boldsymbol{x}_{3_{[A_b,A_d]}}^c \mid \boldsymbol{y}_{[A_b,A_d]}, \boldsymbol{e}_{[A_b,A_d]}, \boldsymbol{x}_{2_{[A_b]}}^c, \boldsymbol{\Theta}^{c'})$ using (16)

    Generate $U \sim U[0,1]$, **if** $(U < r_{\boldsymbol{x}_3})$ $\{\boldsymbol{x}_{3_{[A_n,A_d]}}^c = \boldsymbol{x}_{3_{[A_n \setminus A_b,A_d]}}^c \cup \boldsymbol{x}_{3_{[A_b,A_d]}}^p\}$ **else** $\{\boldsymbol{x}_{3_{[A_n,A_d]}}^c = \boldsymbol{x}_{3_{[A_n,A_d]}}^c\}$

    if $((i \times N_{\text{MH}} < N_{\text{burn}})$ & $(i \times N_{\text{MH}} \in S_6))$ {use adaptive strategy to change the step size parameter} **else** {no change}

  end for

  for $k = 1$ to 4 **do**

    Metropolis–Hastings correction for $\boldsymbol{\Theta}_k$: Calculate the acceptance ratio $r_{\boldsymbol{\Theta}_k} = R_{\boldsymbol{\Theta}_k}(\boldsymbol{\Theta}_k^c, \boldsymbol{\Theta}_k^{c'} \mid \boldsymbol{y}_{[A_n,A_d]}, \boldsymbol{e}_{[A_n,A_d]}, \boldsymbol{x}_{2_{[A_n]}}^c, \boldsymbol{x}_{3_{[A_n,A_d]}}^c, \boldsymbol{\Theta}_{-k}^{c'})$ using (14)

    Generate $U \sim U[0,1]$, **if** $(U < r_{\boldsymbol{\Theta}_k})$ $\{\boldsymbol{\Theta}_k^{c'} = \boldsymbol{\Theta}_k^c,\ \boldsymbol{\Theta}_k^c = \boldsymbol{\Theta}_k^c\}$ **else** $\{\boldsymbol{\Theta}_k^{c'} = \boldsymbol{\Theta}_k^{c'},\ \boldsymbol{\Theta}_k^c = \boldsymbol{\Theta}_k^{c'}\}$

    if $((i \times N_{\text{MH}} < N_{\text{burn}})$ & $(i \times N_{\text{MH}} \in S_k))$ {use adaptive strategy to change the step size parameter} **else** {no change}

  end for

end for

where $\boldsymbol{y}_b$ is the vector of length b sampled without replacement from the full data vector $\boldsymbol{y}_n$, and $\frac{n}{b}\nabla^{\star}_{\boldsymbol{z}} \log \pi(\boldsymbol{z} \mid \boldsymbol{y}_b)$ is an unbiased estimator of $\nabla_{\boldsymbol{z}} \log \pi(\boldsymbol{z} \mid \boldsymbol{y}_n)$, based on $\boldsymbol{y}_b$ only (i.e., it is computed from the summands in the gradient corresponding to $\boldsymbol{y}_b$, rescaled by n/b). It can be shown that the SGLD algorithm (13) has theoretical guarantees to converge to the exact stationary distribution as the step size $\tau \equiv \tau_s \to 0$, such that $\sum \tau_s = \infty$ and $\sum \tau_s^2 < \infty$, where τ_s denote the step size at s^{th} iteration; see Welling and Teh (2011) for more details. In practice, it is difficult to choose the optimal τ_s parameter as it relies on a bias variance trade-off, and is often chosen by cross-validation in the machine learning literature. As it is difficult to choose τ_s optimally in our complex models, we instead rely on Metropolis–Hastings corrections applied after a fixed number of iterations. Here, we propose two different SGLD-based MCMC algorithms (i.e., Algorithm 1 and a simpler version thereof, Algorithm 2, described in the Supplementary Material) and we study their performance by simulation in Section 3.4 and in the Supplementary Material. Note that, when the SGLD algorithm (13) is applied to the vector of latent variables in our model, we update only a subvector of the latent variables and therefore the rescaling factor n/b in (13) is not necessary. Furthermore, after Metropolis–Hastings correction we may interpret it as a block MALA sampling scheme.

For conciseness, we here report only the details of Algorithm 1, which is based on the SGLD algorithm with Metropolis–Hastings corrections, as it turns out to be the best among the two proposed algorithms in terms of accuracy and computational cost. For details about Algorithm 2, see Section 2 of the Supplementary Material.

Algorithm 1 is based on the SGLD algorithm and consists of three successive steps. Let the whole hyperparameter vector $\boldsymbol{\Theta}$ be divided into four blocks as $\boldsymbol{\Theta} = (\boldsymbol{\Theta}_1^T, \boldsymbol{\Theta}_2^T, \boldsymbol{\Theta}_3^T, \boldsymbol{\Theta}_4^T)^T$, where $\boldsymbol{\Theta}_1 = (\gamma_0, \gamma_1, \ldots, \gamma_p)^T$, $\boldsymbol{\Theta}_2 = \beta_1$, $\boldsymbol{\Theta}_3 = \beta_2$, and $\boldsymbol{\Theta}_4 = (\beta_3, \rho)^T$, for the specific model proposed in Section 2.3. In the first step, we update the hyperparameters $\boldsymbol{\Theta}_1$, $\boldsymbol{\Theta}_2$, $\boldsymbol{\Theta}_3$, and $\boldsymbol{\Theta}_4$ in four different blocks using the SGLD algorithm (13) with a random sub-dataset of size b. At every MCMC iteration, we propose a new state using the SGLD algorithm (13) and accept it with probability one, and

then at the end of every fixed number (N_{MH}) of iterations we either accept or reject the whole trajectory using the standard Metropolis–Hastings criterion; see the pseudo-code in Algorithm 1 for more details. In the second and third steps, we update a random subset of size b of the latent parameters $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$, respectively, through the SGLD proposal (13), which is the same as using exact gradient in (12), and after applying the Metropolis–Hastings correction it becomes a block MALA algorithm; see the pseudo-code in Algorithm 1 for more details. Denote $q_1(\boldsymbol{\Theta}_k^p \mid \boldsymbol{\Theta}_k)$, $q_2(\boldsymbol{x}_2^p \mid \boldsymbol{x}_2)$, and $q_3(\boldsymbol{x}_3^p \mid \boldsymbol{x}_3)$ the proposal distributions for the hyperparameter vector $\boldsymbol{\Theta}_k$, the latent parameter vector $\boldsymbol{x}_2$, and the other latent parameter vector $\boldsymbol{x}_3$, respectively, where the superscript p refers to proposed values. Let $\alpha_{\boldsymbol{\Theta}_k}(\boldsymbol{\Theta}_k^p, \boldsymbol{\Theta}_k)$, $\alpha_{\boldsymbol{x}_2}(\boldsymbol{x}_2^p, \boldsymbol{x}_2)$, and $\alpha_{\boldsymbol{x}_3}(\boldsymbol{x}_3^p, \boldsymbol{x}_3)$ denote the acceptance probabilities for $\boldsymbol{\Theta}_k$, $\boldsymbol{x}_2$, and $\boldsymbol{x}_3$, respectively. Then, $\alpha_{\boldsymbol{\Theta}_k}(\boldsymbol{\Theta}_k^p, \boldsymbol{\Theta}_k) = \min\left\{1, R_{\boldsymbol{\Theta}_k}(\boldsymbol{\Theta}_k^p, \boldsymbol{\Theta}_k \mid \boldsymbol{y}, \boldsymbol{e}, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{\Theta}_{-k})\right\}$, $\alpha_{\boldsymbol{x}_2}(\boldsymbol{x}_2^p, \boldsymbol{x}_2) = \min\left\{1, R_{\boldsymbol{x}_2}(\boldsymbol{x}_2^p, \boldsymbol{x}_2 \mid \boldsymbol{y}, \boldsymbol{e}, \boldsymbol{\Theta}, \boldsymbol{x}_3)\right\}$, $\alpha_{\boldsymbol{x}_3}(\boldsymbol{x}_3^p, \boldsymbol{x}_3) = \min\left\{1, R_{\boldsymbol{x}_3}(\boldsymbol{x}_3^p, \boldsymbol{x}_3 \mid \boldsymbol{y}, \boldsymbol{e}, \boldsymbol{x}_2, \boldsymbol{\Theta})\right\}$, where the corresponding acceptance ratios are

$$R_{\boldsymbol{\Theta}_k}(\boldsymbol{\Theta}_k^p, \boldsymbol{\Theta}_k \mid \boldsymbol{y}, \boldsymbol{e}, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{\Theta}_{-k}) = \frac{L\left\{(\boldsymbol{\Theta}_k^p, \boldsymbol{\Theta}_{-k}), \boldsymbol{x}_2, \boldsymbol{x}_3; \boldsymbol{y}, \boldsymbol{e}\right\} \pi(\boldsymbol{\Theta}_k^p)\, q_1(\boldsymbol{\Theta}_k \mid \boldsymbol{\Theta}_k^p)}{L\{(\boldsymbol{\Theta}_k, \boldsymbol{\Theta}_{-k}), \boldsymbol{x}_2, \boldsymbol{x}_3; \boldsymbol{y}, \boldsymbol{e}\}\pi(\boldsymbol{\Theta}_k)\, q_1(\boldsymbol{\Theta}_k^p \mid \boldsymbol{\Theta}_k)}, \quad k = 1, 2, 3, 4, \tag{14}$$

$$R_{\boldsymbol{x}_2}(\boldsymbol{x}_2^p, \boldsymbol{x}_2 \mid \boldsymbol{y}, \boldsymbol{e}, \boldsymbol{\Theta}, \boldsymbol{x}_3) = \frac{L(\boldsymbol{\Theta}, \boldsymbol{x}_2^p, \boldsymbol{x}_3; \boldsymbol{y}, \boldsymbol{e})\, q_2(\boldsymbol{x}_2 \mid \boldsymbol{x}_2^p)}{L(\boldsymbol{\Theta}, \boldsymbol{x}_2, \boldsymbol{x}_3; \boldsymbol{y}, \boldsymbol{e})\, q_2(\boldsymbol{x}_2^p \mid \boldsymbol{x}_2)}, \tag{15}$$

$$R_{\boldsymbol{x}_3}(\boldsymbol{x}_3^p, \boldsymbol{x}_3 \mid \boldsymbol{y}, \boldsymbol{e}, \boldsymbol{x}_2, \boldsymbol{\Theta}) = \frac{L(\boldsymbol{\Theta}, \boldsymbol{x}_2, \boldsymbol{x}_3^p; \boldsymbol{y}, \boldsymbol{e})\, q_3(\boldsymbol{x}_3 \mid \boldsymbol{x}_3^p)}{L(\boldsymbol{\Theta}, \boldsymbol{x}_2, \boldsymbol{x}_3; \boldsymbol{y}, \boldsymbol{e})\, q_3(\boldsymbol{x}_3^p \mid \boldsymbol{x}_3)}, \tag{16}$$

with $L(\boldsymbol{\Theta}, \boldsymbol{x}_2, \boldsymbol{x}_3; \boldsymbol{y}, \boldsymbol{e})$ the censored likelihood in (9), and $\boldsymbol{\Theta}_{-k}$ denoting the hyperparameter vector after removing the k^{th} block from the full hyperparameter vector $\boldsymbol{\Theta}$. The step size parameter τ in (13) determines the performance of the MCMC sampler and is directly responsible for jump sizes in the chains of $\{\boldsymbol{\Theta}_k, k = 1, 2, 3, 4\}$, $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$ at each MCMC iteration. Here, we have in fact six step size parameters to set, namely, $\{\tau_{\boldsymbol{\Theta}_k}, k = 1, 2, 3, 4\}$, $\tau_{\boldsymbol{x}_2}$ and $\tau_{\boldsymbol{x}_3}$, corresponding to the SGLD updates for $\boldsymbol{\Theta}_k$'s, $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$, respectively. We select them adaptively, during the burn-in phase of the MCMC algorithm, in order to achieve a desired acceptance probability. Let τ denote a generic step size parameter. We tune τ using the adaptive algorithm from Yadav et al.

(2021): during an initial burn-in phase we modify the current value, τ_{cur}, of τ every 500 iterations as $\tau_{\text{cur}} \mapsto \tau_{\text{new}} := \exp\{(P_{\text{cur}} - P_{\text{tar}})/\theta\}\,\tau_{\text{cur}}$, where P_{tar} is the target acceptance probability, P_{cur} is the current acceptance probability (computed from the last 500 iterations), and $\theta > 0$ is a parameter modulating the rate of change of τ. Here, we set $P_{\text{tar}} = 0.23$ for random walk proposals and $P_{\text{tar}} = 0.57$ for SGLD-based proposals. In a second burn-in phase, we update the tuning parameters using the same adaptive strategy only if the acceptance probability drops out of the intervals $[0.15, 0.30]$ and $[0.50, 0.65]$ for SGLD-based proposals and block MALA proposals, respectively.

Since all proposals are based on the Gaussian distribution, we transform the parameters so that their support becomes the whole real line through the following reparametrization:

$$\tilde{\gamma}_l = \gamma_l,\ l = 0, 1, \ldots, p,\ \tilde{\beta}_1 = \log\left(\frac{\beta_1}{\delta_1 - \beta_1}\right),\ \tilde{\beta}_2 = \log\left(\frac{\beta_2}{\delta_2 - \beta_2}\right),\ \tilde{\beta}_3 = -\log(\beta_3 - 1),\ \tilde{\rho} = \log\left(\frac{\rho}{2\delta - \rho}\right).$$

where δ_1 and δ_2 are the upper limits for the β_1 and β_2 parameters, respectively, often set to some finite positive values (e.g., $\delta_1 = \delta_2 = 1$ or 2) to avoid numerical issues, and δ is the maximum distance between the stations (i.e., the "diameter" of the study region). The corresponding reverse transformation is

$$\gamma_l = \tilde{\gamma}_l,\ l = 0, 1, \ldots, p,\ \beta_1 = \frac{\delta_1 \exp(\tilde{\beta}_1)}{1 + \exp(\tilde{\beta}_1)},\ \beta_2 = \frac{\delta_2 \exp(\tilde{\beta}_2)}{1 + \exp(\tilde{\beta}_2)},\ \beta_3 = 1 + \exp(-\tilde{\beta}_3),\ \rho = \frac{2\delta \exp(\tilde{\rho})}{1 + \exp(\tilde{\rho})}.$$

The Jacobian matrix of the transformation is $\log(J) = \log(\delta_1) + \tilde{\beta}_1 - 2\log\{1 + \exp(\tilde{\beta}_1)\} + \log(\delta_2) + \tilde{\beta}_2 - 2\log\{1 + \exp(\tilde{\beta}_2)\} - \tilde{\beta}_3 + \log(2) + \log(\delta) + \tilde{\rho} - 2\log\{1 + \exp(\tilde{\rho})\}$. Similarly, we log-transform the latent parameters as $\tilde{\boldsymbol{x}}_2 = \log(\boldsymbol{x}_2)$ and $\tilde{\boldsymbol{x}}_3 = \log(\boldsymbol{x}_3)$.

3.4 Simulation study

We now check the performance of our SGLD-based MCMC sampler (Algorithm 1) by simulation with respect to different batch sizes b. We also provide a simulation study for the comparison between the Algorithm 1 and Algorithm 2; for conciseness, the results are reported in the Supplementary Material. We simulate data from the spatial product mixture model of the form (3),

written in hierarchical form in (4), and further specified in (7) (recall Section 2.3), for $d = 100$ spatial locations and $n = 200$ time replicates (i.e., for a total of $100 \times 200 + 200 = 20,200$ latent variables). The spatial locations are generated uniformly in the unit square $[0,1]^2$ and the latent term $\boldsymbol{X}_{3t}$ has an underlying Gaussian copula with a stationary isotropic exponential correlation function $\sigma(h) = \exp(-\|h\|/\rho)$, where $\rho > 0$ is the range parameter, set here to $\rho = 0.5$. For the purpose of spatial prediction, we completely mask the data at 20 sites, such that their simulated values are treated as fully missing. The censoring threshold u_{tj} is here fixed to the site-wise 75% quantile based on the observations $(y_{1j}, \ldots, y_{nj})^T$, and we set $u_{tj} = \infty$, whenever data y_{tj} is missing. The parameters are chosen as $\beta_1 = 0.8$, $\beta_2 = 0.7$, $\beta_3 = 5$ (i.e., tail index $\xi = 0.2$), and the scale parameter is modeled spatially as $\boldsymbol{\alpha} = \exp\big(\gamma_0 \mathbf{1} + \gamma_1 \boldsymbol{Z}_1 + \gamma_2 \boldsymbol{Z}_2 + \gamma_3 \boldsymbol{Z}_3\big)$, where $\exp(\gamma_0) = \gamma_1 = \gamma_2 = \gamma_3 = 1$, $\boldsymbol{Z}_1$ denotes the x-coordinate of each site, $\boldsymbol{Z}_2$ denotes the y-coordinate of each site, and $\boldsymbol{Z}_3$ is a covariate that is randomly generated from the standard normal distribution.

To make inference, we use the MCMC sampler detailed in Algorithm 1, where we set $N_{\text{MH}} = 25$, $\delta_1 = \delta_2 = 1$, so that the parameters β_1 and β_2 lie within the interval $[0,1]$, and δ is set to the maximum spatial distance between the observed locations. We run two MCMC chains in parallel with two different initial values to check the convergence of Markov chains, and we compute the posterior summaries by averaging post-burn-in values from the two Markov chains.

Figure 2 shows the trace plots for all hyperparameters for two different batch sizes b. The first two rows show the Markov chains for Algorithm 1 with batch size $b = 5$, and the last two rows correspond to batch size $b = 20$. For both batch sizes, the MCMC chains show good and comparable mixing performance and relatively fast convergence, especially for the regression parameters $\gamma_0, \gamma_1, \ldots, \gamma_p$, as well as for the "weights" β_1 and β_2. The parameters β_3 and ρ are slower to converge but they appear to suitably converge after about 0.5 million iterations. The true values for all hyperparameters are close to the posterior means with a relatively narrow 95% credible interval, suggesting that the MCMC algorithm performs well overall. The run-time is almost 9.5

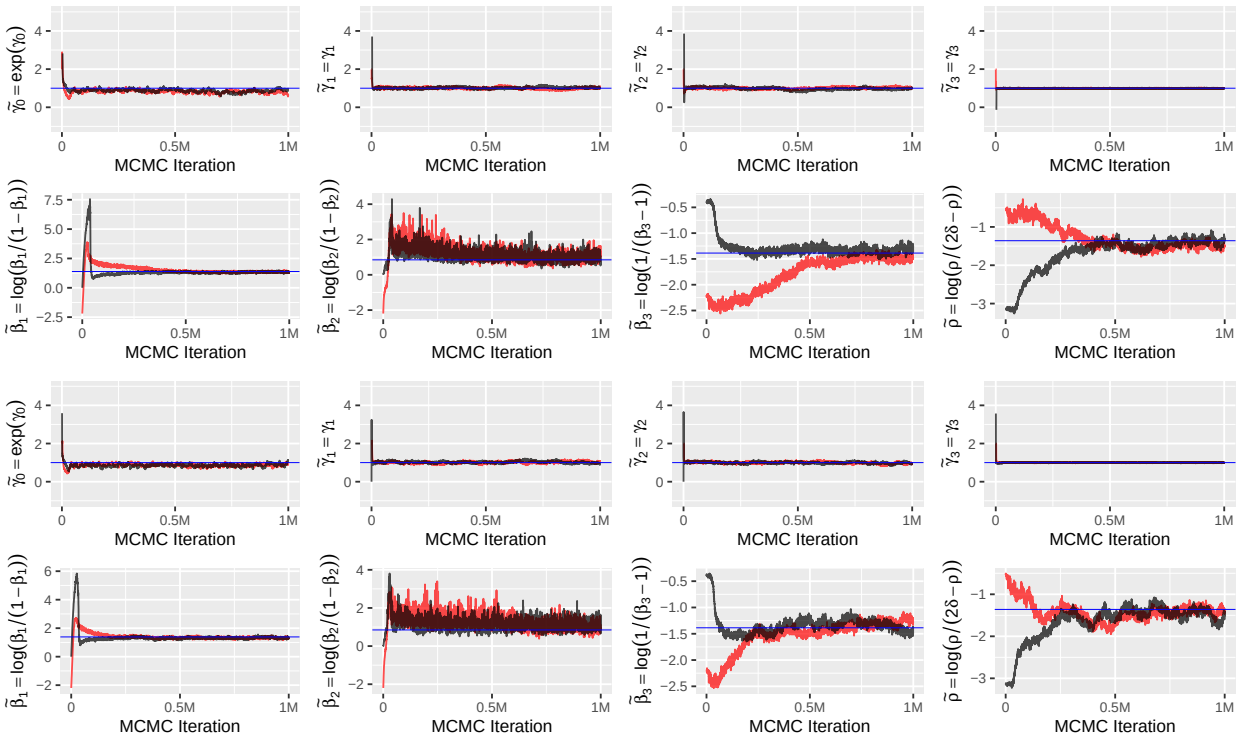

Figure 2: Trace plots for all hyperparameters in our simulation study. The first two rows correspond to the Algorithm 1 with batch size $b = 5$, and the last two rows are with batch size $b = 20$. The two Markov chains (red and black), correspond to two different initial values. The total number of MCMC sample iterations is 1 million (1M). The blue horizontal lines show the true values.

hours to run 1 million iterations when the batch size is set to $b = 5$, and it takes about 14 hours to run 1 million iterations when the batch size is set to $b = 20$. Significant speed-up can thus be obtained without compromising much on the convergence of Markov chains. Table 1 compares the performance of the SGLD-based MCMC algorithm (Algorithm 1) with respect to different batch sizes, $b = 5, 10$ and 20. The results are similar for different batch sizes, so in our application we choose the batch size $b = 5$ as it yields significant speed-up and the highest effective sample size per minute (ESS/min) for the hyperparameters that seem to be the most tricky to estimate and have the slowest convergence rate based on Figure 2 (i.e., β_1, β_3 and ρ).

In Figure 3, we examine the predictive performance of our algorithm at the 20 unobserved sites (with masked observations) by comparing boxplots of the true (masked) observations to samples from the corresponding posterior predictive distribution. Samples from the posterior predictive

Table 1: Absolute bias, standard error, 95% credible interval (CI) length, effective sample size per minute (ESS/min.) for the SGLD-based MCMC algorithm (Algorithm 1) for different batch sizes $b = 5, 10, 20$ in our simulation study. All posterior summary statistics are calculated after removing the first $3N_{tot}/4$ burn-in samples, where $N_{tot} = 1$ million is the total number of MCMC iterations.

	batch size b	$\exp(\gamma_0)$	γ_1	γ_2	γ_3	β_1	β_2	β_3	ρ
Absolute bias	5	0.18	0.01	0	0.01	0.01	0.02	0.08	0.03
	10	0.09	0.02	0.04	0.01	0.01	0.07	0.13	0.04
	20	0.13	0.03	0.03	0.01	0.01	0.05	0.24	0.03
Standard error	5	0.06	0.03	0.03	0.01	0.01	0.03	0.16	0.02
	10	0.05	0.03	0.04	0.01	0.01	0.03	0.22	0.03
	20	0.05	0.03	0.04	0.01	0.01	0.03	0.16	0.03
95% CI length	5	0.23	0.12	0.1	0.04	0.02	0.13	0.61	0.08
	10	0.18	0.12	0.13	0.03	0.02	0.12	0.83	0.12
	20	0.2	0.12	0.14	0.03	0.03	0.12	0.65	0.12
ESS/min.	5	2.96	1.79	7.2	177.29	23.97	54.78	13.88	4.75
	10	9.66	1.69	2.25	150.67	16.43	57.35	2.41	2.23
	20	13.97	4.04	1.8	114.87	5.8	41.4	4.38	2.8

distribution are obtained using the product mixture construction (3), where the model hyperparameters are estimated using the sample posterior median. The posterior predictive boxplots are similar to the boxplots of the true data, and capture the variability at each site quite well. This suggests that our algorithm succeeds in performing spatial prediction. Thanks to our model construction, it is possible to use our censored inference approach to simultaneously fit the model to high threshold exceedances and perform spatial prediction at unobserved locations efficiently.

4 Application to extreme precipitation data in Spain

4.1 Data description

To illustrate our methodology, we now study precipitation intensities from Spain, publicly available from the European Climate Assessment & Dataset project (see link). The dataset reports daily precipitation amounts observed at more than 200 spatial locations from 1941 to 2018. We apply our spatial product mixture model to a subset corresponding to a study region in North-Eastern Spain with $d = 94$ observation sites, and we consider the study period from 2011 to 2020. To avoid modeling complex temporal nonstationarites, we keep observations from September to December (i.e., the most rainy season), resulting in $n = 1220$ temporal replicates (for a total of $94 \times 1220 +$

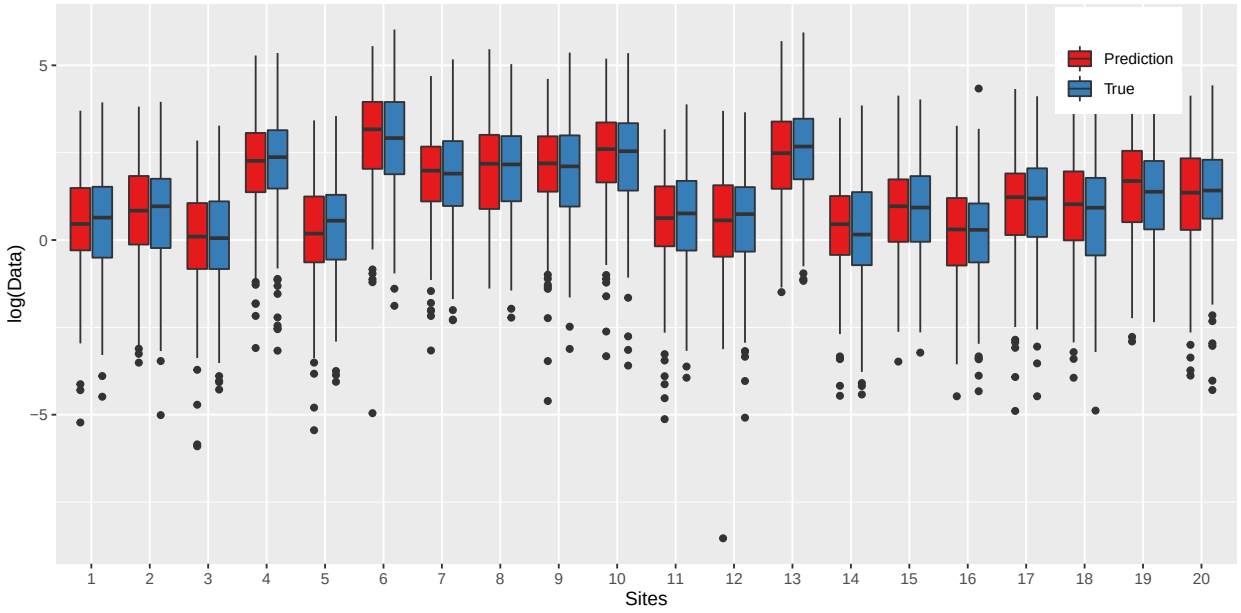

Figure 3: Boxplots of samples from the predictive distribution (red) at the 20 unobserved sites, along with boxplots of true (masked) observations at the same sites (blue), both in log scale. Here, the batch size of Algorithm 1 is set to $b = 5$.

$1220 = 115,900$ latent variables). The distance between the two furthest sites is 260km, and the two closest sites are 0.17km apart. The site-wise proportion of missing observations varies from 0.01% to 100%. Also, the proportion of zeros at each site ranges from 41% to 85% with an average of 70%. The left panel in Figure 4 shows the site-wise mean precipitation plot, and the right panel shows the pairwise tail correlation plot (i.e., the $\chi(u)$ coefficient) at a fixed threshold $u = 95\%$. These plots show that there is quite strong spatial heterogeneity and strong tail dependence in the data. The northern region receives higher precipitation than the southern region, and the sites at higher altitude receive higher precipitation, as well. This motivates including geographical information such as latitude, longitude and altitude, as covariates in the model.

4.2 Modeling precipitation intensities using spatial product mixture models

We fit our specific spatial product mixture model (7) and four nested sub-models to the Spanish precipitation data. More precisely, we consider the following sub-models:

D1 General product mixture model (3), defined as $\boldsymbol{Y}_t = \boldsymbol{\alpha}\boldsymbol{X}_{1t}\boldsymbol{X}_{2t}\boldsymbol{X}_{3t}$, $t = 1, \ldots, n,$, where

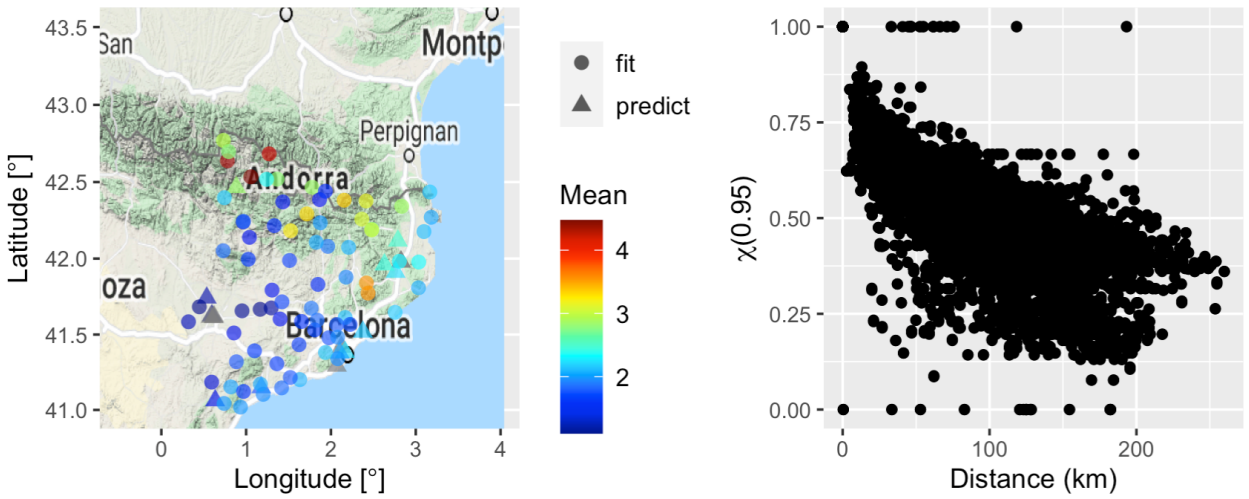

Figure 4: Left: Color-coded mean precipitation amount [mm] at the study sites. Dots represent sites used for fitting and triangles represent sites where spatial prediction is performed. Right: $\chi(u)$ plot at a fixed threshold $u = 0.95$.

marginals F_1, F_2, and F_3 are given by (7).

D2 Model D1 but with $\beta_1 \to 0$ and $\beta_2 \to 0$ in (7), which may be written in the form $\boldsymbol{Y}_t = \boldsymbol{\alpha}\boldsymbol{X}_{3t}$, $t = 1, \ldots, n$.

D3 Model D1 but with $\beta_1 \to 0$ in (7), which may be written as the product mixture model of the form $\boldsymbol{Y}_t = \boldsymbol{\alpha}\boldsymbol{X}_{2t}\boldsymbol{X}_{3t}$, $t = 1, \ldots, n$.

D4 Model D1 but with $\beta_2 \to 0$ in (7), which may be written as the product mixture model of the form $\boldsymbol{Y}_t = \boldsymbol{\alpha}\boldsymbol{X}_{1t}\boldsymbol{X}_{3t}$, $t = 1, \ldots, n$.

Model D1 is the most general model that we have discussed in detail in Section 2.3. The other models, D2, D3, and D4, are sub-models nested within model D1. The β_1 and β_2 parameters act as weight parameters for the fully dependent ($\boldsymbol{X}_2$) and iid ($\boldsymbol{X}_1$) components, respectively, with $\beta_1 \to 0$ and $\beta_2 \to 0$ resulting in degeneration of the corresponding components at one; see Section 2.3 for more details. Therefore, model D2 may be obtained by taking both $\beta_1 \to 0$ and $\beta_2 \to 0$, whereas model D3 and D4 may be obtained by letting either $\beta_1 \to 0$ or $\beta_2 \to 0$, respectively. In our MCMC algorithm, to preserve computational benefits of the three-component formulation, we set

Table 2: Mean absolute error (Abs. Error) and continuous ranked probability score (CRPS) averaged over the 18 prediction stations. Lower values of Abs. Error and CRPS indicate better models, and the bold-blue-colored digits show the best performance among different covariate combinations (M1–M10) and different models (D1–D4), while the light-blue digits show the best performance for a particular choice of covariate combination (M1–M10).

		M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
Abs. Error	D1	40.663	39.422	38.592	40.451	38.975	**38.182**	39.698	40.504	39.935	40.050
	D2	46.231	44.145	43.929	44.866	44.120	44.324	44.946	45.853	45.753	45.320
	D3	43.680	41.698	41.368	42.952	38.887	39.351	40.226	40.090	40.869	40.182
	D4	44.056	42.187	42.741	42.521	42.371	43.080	43.247	44.186	43.655	43.572
CRPS	D1	48.770	44.875	42.913	45.345	43.815	**41.965**	45.802	48.465	47.028	47.223
	D2	115.091	102.041	101.134	99.708	99.069	100.340	103.212	108.340	107.771	105.724
	D3	64.891	56.330	55.457	56.666	47.271	48.651	51.083	51.102	53.355	51.238
	D4	92.834	84.609	86.725	82.864	85.889	89.225	89.932	95.216	93.280	92.533

$\beta_1 = 0.01$ and/or $\beta_2 = 0.01$, whenever $\beta_1 \to 0$ and/or $\beta_2 \to 0$. For each of the models D1–D4, we consider 10 different combinations of covariates in the scale parameter $\boldsymbol{\alpha}$, denoted by models M1–M10, resulting in 40 different product mixture models in total. Precisely, we consider several combinations where the scale vector $\boldsymbol{\alpha}$ comprises linear or quadratic spatial covariates in terms of latitude ($\boldsymbol{Z}_1$), longitude ($\boldsymbol{Z}_2$), and altitude ($\boldsymbol{Z}_3$), and where covariates are standardized to have mean zero and unit variance. We specifically consider the following combinations:

M1: $\boldsymbol{\alpha} = \exp(\gamma_0 \mathbf{1}_d)$;

M2: $\boldsymbol{\alpha} = \exp(\gamma_0 \mathbf{1}_d + \gamma_{\text{lat}} \mathbf{Z}_1)$;

M3: $\boldsymbol{\alpha} = \exp(\gamma_0 \mathbf{1}_d + \gamma_{\text{lat}} \mathbf{Z}_1 + \gamma_{\text{long}} \mathbf{Z}_2)$;

M4: $\boldsymbol{\alpha} = \exp(\gamma_0 \mathbf{1}_d + \gamma_{\text{lat}} \mathbf{Z}_1 + \gamma_{\text{long}} \mathbf{Z}_2 + \gamma_{\text{alt}} \mathbf{Z}_3)$;

M5: $\boldsymbol{\alpha} = \exp(\gamma_0 \mathbf{1}_d + \gamma_{\text{lat}} \mathbf{Z}_1 + \gamma_{\text{long}} \mathbf{Z}_2 + \gamma_{\text{alt}} \mathbf{Z}_3 + \gamma_{\text{lat}^2} \mathbf{Z}_1^2)$;

M6: $\boldsymbol{\alpha} = \exp(\gamma_0 \mathbf{1}_d + \gamma_{\text{lat}} \mathbf{Z}_1 + \gamma_{\text{long}} \mathbf{Z}_2 + \gamma_{\text{alt}} \mathbf{Z}_3 + \gamma_{\text{lat}^2} \mathbf{Z}_1^2 + \gamma_{\text{long}^2} \mathbf{Z}_2^2)$;

M7: $\boldsymbol{\alpha} = \exp(\gamma_0 \mathbf{1}_d + \gamma_{\text{lat}} \mathbf{Z}_1 + \gamma_{\text{long}} \mathbf{Z}_2 + \gamma_{\text{alt}} \mathbf{Z}_3 + \gamma_{\text{lat}^2} \mathbf{Z}_1^2 + \gamma_{\text{long}^2} \mathbf{Z}_2^2 + \gamma_{\text{alt}^2} \mathbf{Z}_3^2)$;

M8: $\boldsymbol{\alpha} = \exp(\gamma_0 \mathbf{1}_d + \gamma_{\text{lat}} \mathbf{Z}_1 + \gamma_{\text{long}} \mathbf{Z}_2 + \gamma_{\text{alt}} \mathbf{Z}_3 + \gamma_{\text{lat}^2} \mathbf{Z}_1^2 + \gamma_{\text{long}^2} \mathbf{Z}_2^2 + \gamma_{\text{alt}^2} \mathbf{Z}_3^2 + \gamma_{\text{lat.long}} \mathbf{Z}_1 \mathbf{Z}_2)$;

M9: $\boldsymbol{\alpha} = \exp(\gamma_0 \mathbf{1}_d + \gamma_{\text{lat}} \mathbf{Z}_1 + \gamma_{\text{long}} \mathbf{Z}_2 + \gamma_{\text{alt}} \mathbf{Z}_3 + \gamma_{\text{lat}^2} \mathbf{Z}_1^2 + \gamma_{\text{long}^2} \mathbf{Z}_2^2 + \gamma_{\text{alt}^2} \mathbf{Z}_3^2 + \gamma_{\text{lat.long}} \mathbf{Z}_1 \mathbf{Z}_2 + \gamma_{\text{long.alt}} \mathbf{Z}_2 \mathbf{Z}_3)$;

M10: $\boldsymbol{\alpha} = \exp(\gamma_0 \mathbf{1}_d + \gamma_{\text{lat}} \mathbf{Z}_1 + \gamma_{\text{long}} \mathbf{Z}_2 + \gamma_{\text{alt}} \mathbf{Z}_3 + \gamma_{\text{lat}^2} \mathbf{Z}_1^2 + \gamma_{\text{long}^2} \mathbf{Z}_2^2 + \gamma_{\text{alt}^2} \mathbf{Z}_3^2 + \gamma_{\text{lat.long}} \mathbf{Z}_1 \mathbf{Z}_2 + \gamma_{\text{long.alt}} \mathbf{Z}_2 \mathbf{Z}_3 + \gamma_{\text{lat.alt}} \mathbf{Z}_1 \mathbf{Z}_3)$.

The censoring threshold for each of the models D1–D4 and covariate combinations M1–M10 is set to the 75% site-wise quantile of the strictly positive precipitation intensities. We choose 76

Table 3: Posterior summary statistics for our best model D1 with covariate combination M6, calculated based on $N_{tot}/4$ samples after deleting the first $3N_{tot}/4$ burn-in samples, where $N_{tot} = 1$M is the total number of MCMC samples.

	$\exp(\gamma_0)$	γ_{lat}	γ_{long}	γ_{alt}	γ_{lat^2}	γ_{long^2}	β_1	β_2	$\xi = 1/\beta_3$	ρ [km]
Post. mean	2.895	0.013	0.215	0.234	0.220	0.030	1.206	1.993	0.098	519.520
Standard dev.	0.055	0.022	0.024	0.016	0.018	0.018	0.011	0.005	0.002	0.028
Lower 95% CI	2.790	-0.029	0.170	0.202	0.187	-0.004	1.185	1.980	0.094	519.458
Upper 95% CI	3.010	0.056	0.264	0.264	0.256	0.064	1.229	1.999	0.102	519.561
ESS/min.	0.472	67.772	86.969	90.176	68.958	97.877	34.852	163.682	0.207	4.444

stations for fitting and leave 18 stations for prediction and model validation, where the true data are masked. For missing observations, we set the censoring threshold to $u_{tj} = \infty$. For inference, we use the SGLD-based scheme (Algorithm 1) presented in Section 3 with one million (1M) iterations, $N_{\text{MH}} = 25$, and with a batch size of $b = 5$ for all the models. Also, we set $\delta_1 = \delta_2 = 2$, to avoid numerical issue and given that the data show strong dependence, so that the β_1 and β_2 parameters lie within the intervals $[0, 2]$.

Table 2 compares the model performances based on the mean absolute error (Abs. Error) and the continuous ranked probability score (CRPS; Gneiting and Raftery, 2007), averaged over the 18 prediction stations. Clearly, the most general product mixture model D1 outperforms the other sub-models D2, D3, and D4 in terms of lower Abs. Error and CRPS. In particular, model D1 with covariate combination M6 provides the best performance for the prediction at unobserved spatial locations. Moreover, the QQ-plots in Figure 5 show that the marginal predictive performance of our model D1 with covariate combination M6 is satisfactory. Thus, we select model D1 with covariate combination M6 as our best model, as it yields the best out-of-sample goodness-of-fit diagnostics, and can adequately capture the complex spatially-varying dynamics of precipitation extremes. The total run-time for this model is approximately 28 hours to generate one million samples. The mean squared prediction error (MPE) and the tail-weighted CRPS (twCRPS; Lerch et al., 2017), averaged over the 18 prediction stations, shown in Table 3 in the Supplementary Material, provide similar interpretations.

Table 3 shows posterior summary statistics for the best model M6. The estimates of covariate

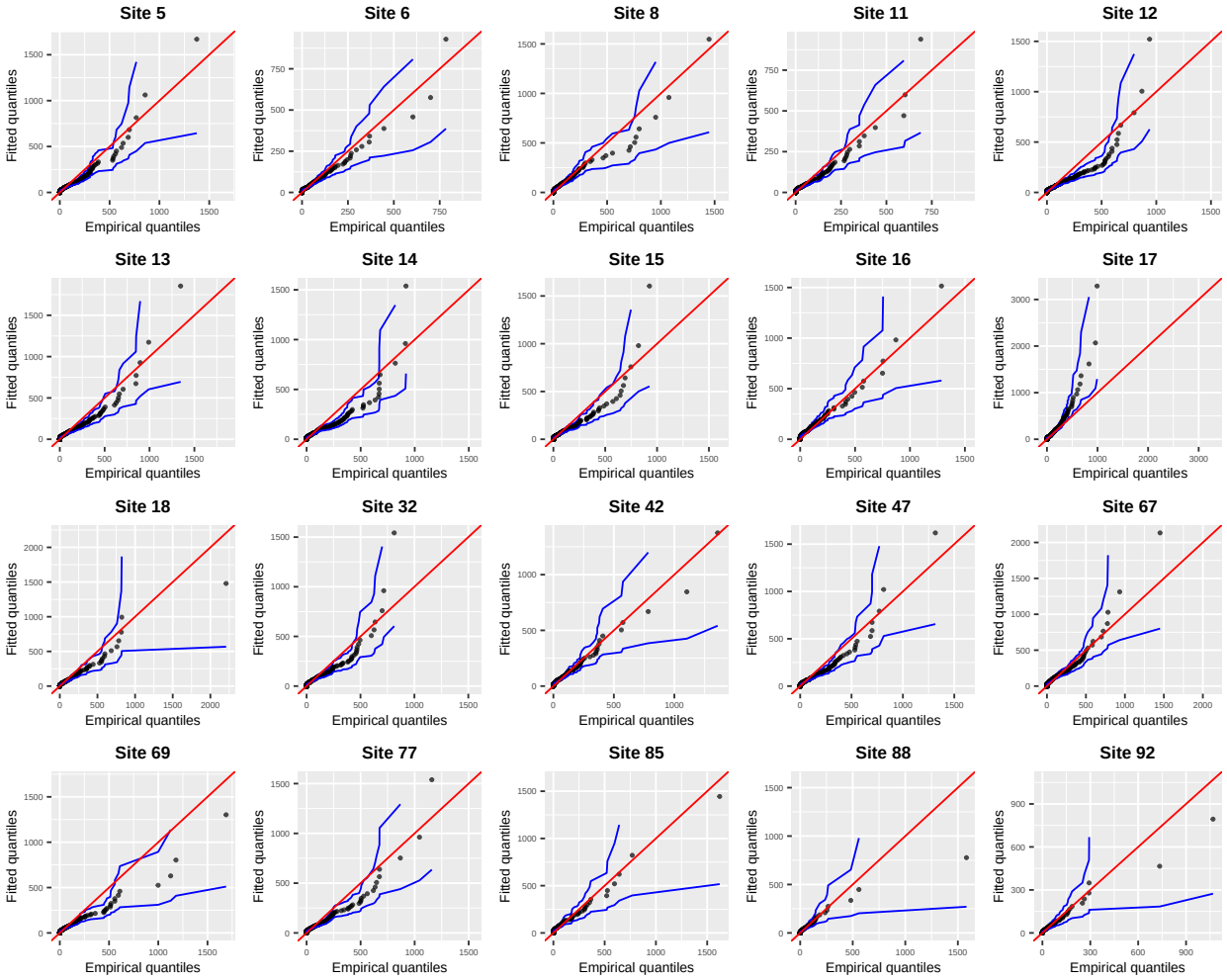

Figure 5: QQ-plots at all prediction stations where the data have been masked (up to site #18 where the precipitation data are available), and some selected fitted sites based on the best model D1 with covariate combination M6. The fitted quantiles are simulated from the specific product model in Section 2.3, where the model hyperparameters are estimated using the corresponding posterior mean, calculated based on $N_{tot}/4$ samples after removing the first $3N_{tot}/4$ burn-in samples, where $N_{tot} = 1\text{M}$ is the total number of MCMC samples.

coefficients such as longitude ($\widehat{\gamma}_{\text{long}} = 0.215$), squared latitude ($\widehat{\gamma}_{\text{lat}^2} = 0.220$) , and altitude ($\widehat{\gamma}_{\text{alt}} = 0.234$), are highly significant, which shows that these geographical covariates are needed in the model. The estimate of β_2 is close to 2, which confirms that there is strong spatial dependence in the data and that the spatially constant term $\boldsymbol{X}_{2t}$ in (3) is crucially needed. The estimate of β_1 is about 1.206, indicating that there is also non-negligible small-scale variability. The estimated tail index is $\widehat{\xi} = 0.098$, indicating moderately heavy tails, which is in line with most precipitation data.

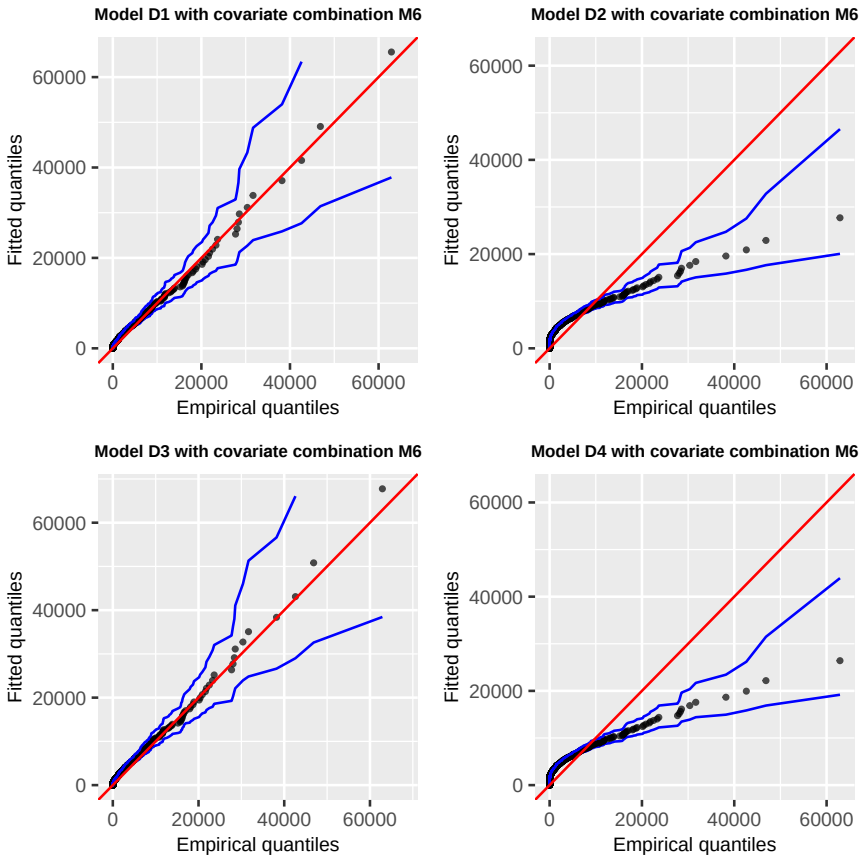

Figure 6: QQ-plot of fitted vs. observed spatial aggregates obtained after summing precipitation amounts over all the 94 spatial locations for model D1 (top left), D2 (top right), D3 (bottom left), and D4 (bottom right), all based on covariate combination M6.

Figure 6 shows the QQ-plots for the spatial aggregates obtained after aggregating over all the 94 spatial locations for each model D1-D4 and covariate combination M6. Precisely, if $Y_t(\boldsymbol{s})$ denotes the precipitation data at spatial location $\boldsymbol{s}$ and time replicate t, then $S_t = \sum_{j=1}^{d} Y_t(\boldsymbol{s}_j)$, $t = 1, \ldots, n$ is the t^{th} spatial aggregate. The QQ-plots in Figure 6 show that models D1 and D3 provide satisfactory performance in predicting the compound risk associated with spatial precipitation aggregates; however, models D2 and D4 strongly underestimate this risk. Note that unlike models D2 and D4, models D1 and D3 involve the fully spatially dependent component, $\boldsymbol{X}_{2t}$. This demonstrates the strong limitation of hierarchical models that are conditionally independent at the data level, and it suggests that it is important to include $\boldsymbol{X}_{2t}$ when modeling spatial precipitation intensities from Eastern Spain. A similar interpretation follows from Tables 2 and 3 in the Supplementary Material.

5 Conclusion

In this paper, we have provided a constructive modeling framework for extreme spatial threshold exceedances based on product mixtures of three distinct and mutually independent random fields, where each of the fields is characterized by a distinct combination of heavy- or lighter-tailed margins and spatial dependence characteristics. These models provide high flexibility in the tail and at sub-asymptotic levels and may be used to capture strong tail dependence in high threshold exceedances. The dependence strength of our proposed model depends on the choice of the underlying copula structure at the latent level, and therefore we can get a variety of flexible dependence structures depending on the latent copula specification.

We here have designed an SGLD-based MCMC algorithm to fit our model efficiently in high spatio-temporal dimensions, where the dimension of the latent parameter vector is comparable to the data dimension. By using the SGLD algorithm, we bypass the expensive calculation of full censored-likelihood, and full gradients, and hence inference is significantly faster with high data dimensions. Thanks to the SGLD algorithm, we can indeed drastically reduce the computational cost of each MCMC iteration, allowing us to fit complex Bayesian hierarchical models with strong data-level tail dependence to threshold exceedances in high dimensions. Our proposed SGLD-based inference is general and can be used to fit several other types of Bayesian hierarchical models, and the scalability of this approach depends on the structure and sophistication of the underlying model. Our proposed general product mixture model is quite complex, with complicated log-likelihood structures and, in particular, the full conditional distribution of the high-dimensional latent parameter vectors $\boldsymbol{X}_2$ and $\boldsymbol{X}_3$ does not have closed-form full conditionals, making it more challenging to fit in high spatio-temporal dimensions. From our experience, based on some simulation studies and data applications, we can handle up to 500 spatial locations with a relatively large number of time replicates, around 1000 for our most general product mixture model. On the contrary, we may extend this to thousands of spatial locations when the underlying models

have a well-behaved structure with nice forms of the log-likelihood, for example, log-Gaussian Cox processes. One particular example of successfully implementing our SGLD algorithm may be found in Cisneros et al. (2021), where log-Gaussian Cox processes are fitted to US wildfire count data with spatial dimension 3503, and 161 independent temporal replicates at each site. One particular limitation behind our SGLD-based MCMC inference scheme is that it requires independent temporal replicates. However, with a slight modification, we could make it work for dependent time replicates as well, similar to the idea proposed in Aicher et al. (2019) that utilizes the information of consecutive time replicates of batch size b, instead of random drawings of time replicates.

In our data application, we have shown how to model precipitation extremes, and have illustrated our methodology on data from North-Eastern Spain. Although our methodology was illustrated with the specific spatial mixture model of Section 2.3, our constructive modeling framework is very general and can lead to several alternative heavy- or light-tailed models (such as those briefly described at the end of Section 2.3), and it would be interesting to explore this in future research. Other research directions include extending our spatial product mixture models to the spatio-temporal context. This may be performed by either incorporating temporal dependence in the fully spatially-dependent latent parameter, $\boldsymbol{X}_{2t}$, or by introducing space-time dependence in the other latent parameter vector, $\boldsymbol{X}_{3t}$. In the space-time context, products of more than three latent processes with distinct spatio-temporal characteristics may also be envisioned (though adding even more complexity), and it would be interesting to investigate how to further generalize our construction to capture different asymptotic dependence regimes in space and time.

Acknowledgments

This publication is based upon work supported by the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research (OSR) under Awards No. OSR-CRG2017-3434 and No. OSR-CRG2020-4394.

References

Aicher, C., Ma, Y.-A., Foti, N. J. and Fox, E. B. (2019) Stochastic gradient mcmc for state space models. SIAM Journal on Mathematics of Data Science **1**(3), 555–587.

Atchadé, Y. F. (2006) An adaptive version for the Metropolis adjusted Langevin algorithm with a truncated drift. Methodology and Computing in applied Probability **8**(2), 235–254.

Bacro, J.-N., Gaetan, C., Opitz, T. and Toulemonde, G. (2020) Hierarchical space-time modeling of asymptotically independent exceedances with an application to precipitation data. Journal of the American Statistical Association **115**(530), 555–569.

Banerjee, S., Carlin, B. P. and Gelfand, A. E. (2014) Hierarchical Modeling and Analysis for Spatial Data. Second edition. CRC Press.

Breiman, L. (1965) On some limit theorems similar to the arc-sin law. Theory of Probability & Its Applications **10**(2), 323–331.

Bücher, A. and Zhou, C. (2021) A horse race between the block maxima method and the peak–over–threshold approach. Statistical Science **36**(3), 360–378.

Castro-Camilo, D., Huser, R. and Rue, H. (2019) A spliced Gamma-generalized Pareto model for short-term extreme wind speed probabilistic forecasting. Journal of Agricultural, Biological and Environmental Statistics **24**(3), 517–534.

Cisneros, D., Gong, Y., Yadav, R., Hazra, A. and Huser, R. (2021) A combined statistical and machine learning approach for spatial prediction of extreme wildfire frequencies and sizes. arXiv preprint arXiv:2112.14920 .

Clark, N. J. and Dixon, P. M. (2021) A class of spatially correlated self-exciting statistical models. Spatial Statistics **43**, 100493.

Cooley, D., Cisewski, J., Erhardt, R. J., Jeon, S., Mannshardt, E., Omolo, B. O. and Sun, Y. (2012) A survey of spatial extremes: Measuring spatial dependence and modeling spatial effects. REVSTAT **10**(1), 135–165.

Cooley, D., Nychka, D. and Naveau, P. (2007) Bayesian spatial modeling of extreme precipitation return levels. Journal of the American Statistical Association **102**(479), 824–840.

Cressie, N. A. C. (1993) Statistics for Spatial Data. Wiley Online Library.

Davison, A. C. and Huser, R. (2015) Statistics of Extremes. Annual Review of Statistics and its Application **2**, 203–235.

Davison, A. C., Huser, R. and Thibaud, E. (2019) Spatial extremes. In Handbook of Environmental and Ecological Statistics, eds A. E. Gelfand, M. Fuentes, J. A. Hoeting and R. L. Smith, pp. 711–744. Boca Raton: CRC press.

Davison, A. C., Padoan, S. and Ribatet, M. (2012) Statistical modelling of spatial extremes (with Discussion). Statistical Science **27**(2), 161–186.

Davison, A. C. and Smith, R. L. (1990) Models for exceedances over high thresholds (with discussion). Journal of the Royal Statistical Society: Series B (Statistical Methodology) **52**(3), 393–442.

Deng, W., Zhang, X., Liang, F. and Lin, G. (2018) Bayesian Deep Learning via stochastic gradient MCMC with a stochastic approximation adaptation. In Proceedings of the International Conference on Learning Representations 2019 Conference, pp. 1–18.

de Fondeville, R. and Davison, A. C. (2018) High-dimensional peaks-over-threshold inference. Biometrika **105**(3), 575–592.

Gneiting, T. and Raftery, A. E. (2007) Strictly proper scoring rules, prediction, and estimation. Journal of the American statistical Association **102**(477), 359–378.

Hazra, A., Huser, R. and Jóhannesson, Á. V. (2021) Bayesian latent Gaussian models for high-dimensional spatial extremes. In Statistical Modeling Using Bayesian Latent Gaussian Models – With Applications in Geophysics and Environmental Sciences, pp. 1–36. Expected to be published by Springer in 2022.

Huser, R., Opitz, T. and Thibaud, E. (2017) Bridging asymptotic independence and dependence in spatial extremes using gaussian scale mixtures. Spatial Statistics **21**, 166–186.

Huser, R. and Wadsworth, J. L. (2019) Modeling spatial processes with unknown extremal dependence class. Journal of the American Statistical Association **114**, 434–444.

Huser, R. and Wadsworth, J. L. (2020) Advances in statistical modeling of spatial extremes. Wiley Interdisciplinary Reviews: Computational Statistics p. e1537.

Johannesson, Á. V., Siegert, S., Huser, R., Bakka, H. and Hrafnkelsson, B. (2021) Approximate Bayesian inference for analysis of spatio-temporal flood frequency data. Annals of Applied Statistics, To appear.

Lerch, S., Thorarinsdottir, T. L., Ravazzolo, F. and Gneiting, T. (2017) Forecaster's dilemma: extreme events and forecast evaluation. Statistical Science **32**(1), 106–127.

Neal, R. M. (2011) MCMC using Hamiltonian dynamics. In Handbook of Markov Chain Monte Carlo, eds G. L. J. S. Brooks, A. Gelman and X.-L. Meng, chapter 5, pp. 113–162. Chapman & Hall/CRC.

Neal, R. M. (2012) Bayesian Learning for Neural Networks. Volume 118. Springer Science & Business Media.

Nemeth, C. and Fearnhead, P. (2020) Stochastic gradient Markov chain Monte Carlo. Journal of the American Statistical Association **116**(533), 433–450.

Opitz, T., Huser, R., Bakka, H. and Rue, H. (2018) INLA goes extreme: Bayesian tail regression for the estimation of high spatio-temporal quantiles. Extremes **21**(3), 441–462.

Resnick, S. I. (1987) Extreme Values, Regular Variation and Point Processes. Springer.

Ribatet, M., Cooley, D. and Davison, A. C. (2012) Bayesian inference from composite likelihoods, with an application to spatial extremes. Statistica Sinica **22**, 813–845.

Roberts, G. O. and Rosenthal, J. S. (1998) Optimal scaling of discrete approximations to Langevin diffusions. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **60**(1), 255–268.

Roberts, G. O. and Tweedie, R. L. (1996) Exponential convergence of Langevin distributions and their discrete approximations. Bernoulli **2**(4), 341–363.

Sang, H. and Gelfand, A. E. (2009) Hierarchical modeling for extreme values observed over space and time. Environmental and Ecological Statistics **16**(3), 407–426.

Sang, H. and Gelfand, A. E. (2010) Continuous spatial process models for spatial extreme values. Journal of Agricultural, Biological, and Environmental Statistics **15**(1), 49–65.

Thibaud, E. and Opitz, T. (2015) Efficient inference and simulation for elliptical Pareto processes. Biometrika **102**(4), 855–870.

Turkman, K. F., Turkman, M. A. and Pereira, J. (2010) Asymptotic models and inference for extremes of spatio-temporal data. Extremes **13**(4), 375–397.

Welling, M. and Teh, Y. W. (2011) Bayesian learning via stochastic gradient Langevin dynamics. In Proceedings of the 28th International Conference on Machine Learning (ICML-11), pp. 681–688.

Yadav, R., Huser, R. and Opitz, T. (2021) Spatial hierarchical modeling of threshold exceedances using rate mixtures. Environmetrics **32**(3), e2662.

Zhang, L., Shaby, B. A. and Wadsworth, J. L. (2021) Hierarchical transformed scale mixtures for flexible modeling of spatial extremes on datasets with many locations. Journal of the American Statistical Association, To appear.

Zhang, R., Li, C., Zhang, J., Chen, C. and Wilson, A. G. (2020) Cyclical stochastic gradient MCMC for Bayesian deep learning. In Proceedings of the International Conference on Learning Representations 2020 Conference, pp. 1–27.