HAL
open science

Multioutput Gaussian processes with functional data: A study on coastal flood hazard assessment

Andrés F. López-Lopera, Déborah Idier, Jeremy Rohmer, François Bachoc

To cite this version:

Andrés F. López-Lopera, Déborah Idier, Jeremy Rohmer, François Bachoc. Multioutput Gaussian processes with functional data: A study on coastal flood hazard assessment. Reliability Engineering and System Safety, 2021, 218, pp.108139. 10.1016/j.ress.2021.108139 . hal-03412417v2

HAL Id: hal-03412417

https://brgm.hal.science/hal-03412417v2

Submitted on 8 Dec 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

MULTIOUTPUT GAUSSIAN PROCESSES WITH FUNCTIONAL DATA: A STUDY ON COASTAL FLOOD HAZARD ASSESSMENT

Andrés F. López-Lopera[*]
andres.lopezlopera@uphf.fr

Déborah Idier[‡]
d.idier@brgm.fr

Jérémy Rohmer[‡]
j.rohmer@brgm.fr

François Bachoc[†]
francois.bachoc@math.univ-toulouse.fr

[*] Université Polytechnique Hauts-de-France, Céramaths, F-59313 Valenciennes, France
[†] Institut de Mathématiques de Toulouse, Université Paul Sabatier, F-31062 Toulouse, France
[‡] BRGM, DRP/R3C, 3 avenue Claude Guillemin, F-45060 Orléans cédex 2, France

ABSTRACT. Surrogate models are often used to replace costly-to-evaluate complex coastal codes to achieve substantial computational savings. In many of those models, the hydrometeorological forcing conditions (inputs) or flood events (outputs) are conveniently parameterized by scalar representations, neglecting that the inputs are actually time series and that floods propagate spatially inland. Both facts are crucial in flood prediction for complex coastal systems. Our aim is to establish a surrogate model that accounts for time-varying inputs and provides information on spatially varying inland flooding. We introduce a multioutput Gaussian process model based on a separable kernel that correlates both functional inputs and spatial locations. Efficient implementations consider tensor-structured computations or sparse-variational approximations. In several experiments, we demonstrate the versatility of the model for both learning maps and inferring unobserved maps, numerically showing the convergence of predictions as the number of learning maps increases. We assess our framework in a coastal flood prediction application. Predictions are obtained with small error values within computation time highly compatible with short-term forecast requirements (on the order of minutes compared to the days required by hydrodynamic simulators). We conclude that our framework is a promising approach for forecast and early-warning systems.

1 Introduction

Natural hazards, such as floods induced by Hurricane Katrina (2005) or by the more recent storms Xynthia (2010) and Johanna (2008), have strong negative impacts on the living conditions of hundreds of people (Blake et al., 2006; Lumbroso and Vinet, 2011; André et al., 2013; Bathrellos et al., 2017). These events caused harmful coastal flooding that resulted in a significant number of deaths. Hurricane Katrina was one of the six most powerful hurricanes ever recorded in the Atlantic. It resulted in a death toll of 1836 and approximately 80 billion dollars of damage (Blake et al., 2006). The storm Xynthia severely impacted low-lying French coastal areas located in the central part of the Bay of Biscay on February 27 and 28, 2010 (Bertin et al., 2012). The flood induced by Xynthia caused 53 fatalities and more than 1 billion euros of material damage. The storm Johanna had minor effects on the French Atlantic coast but still led to significant flooding damage, such as in the town of Gâvres (Britany; André et al., 2013; Idier et al., 2020b). These historical flood episodes reflect the need for accurate forecast and early-warning systems (FEWSs) to reduce the loss of human life and damage in areas at risk of flooding (André et al., 2013; Hoggart et al., 2014; Idier et al., 2020b).

Most of the existing coastal flood FEWSs do not model floods and rely on the prediction of hydrodynamic conditions on the coast and expert judgment (see, e.g., Doong et al., 2012). Some systems under development rely on simplified inland flood computations (see, e.g., Tromble et al., 2019; Stansby et al., 2013), expecting that high-performance computing will allow their integration in operational platforms. In the Netherlands, the operational FEWS issues warnings based on the forecast of the near-shore conditions (like many other FEWSs), but in the case of warnings, floods are estimated using a database of precomputed flooding scenarios. These scenarios were generated for a limited set of dike breach locations and water levels using a model resolution of approximately 10 m. Most of the recent contributions allow modeling of high-resolution floods, even if wave overtopping plays an important role (see, e.g., Le Roy et al., 2015; Idier et al., 2020b). However, those models are costly to evaluate, requiring up to days of parallel computations, making their use in warning forecasting systems impractical.

To overcome the computational complexity of coastal flooding codes, data-driven surrogate models have been widely explored (Sacks et al., 1989; Rohmer et al., 2016; Liu and Guillas, 2017; Li et al., 2020; Bolle et al., 2018; Rueda et al., 2019). These models are initially fed with a statistically rich but tractable number of simulations of a former model. By learning statistical features, surrogate models are then used to predict floods based on knowledge of the coastal state. As shown in (Rohmer and Idier, 2012; Jia and Taflanidis, 2013; Liu and Guillas, 2017; Azzimonti et al., 2019; Betancourt et al., 2020; Perrin et al., 2021), Gaussian process (GP)-based stochastic surrogates can be successfully applied in a wide range of coastal engineering applications. More precisely, they are often used as surrogate models for expensive codes that can be later exploited, such as in regression and sensitivity analysis, both of which are of great interest for reliability engineering and system safety (Betancourt et al., 2020; Perrin et al., 2021; Cao et al., 2021; Da Veiga and Marrel, 2020; Li et al., 2020; Iooss and Ribatet, 2009; Marrel et al., 2009). In (Rohmer and Idier, 2012; Azzimonti et al., 2019), GP-based approaches are used to assess the impact of critical spatial offshore conditions. Both frameworks address level set estimation problems aiming to identify when the offshore conditions exceed a fixed risk threshold. In (Jia and Taflanidis, 2013; Liu and Guillas, 2017), the authors focused on the dimension reduction of expensive computer codes via GP emulators for tsunamis and storm/hurricane risk assessment, respectively. In (Li et al., 2020), a GP-based sensitivity analysis study in the context of expensive models with high-dimensional outputs is conducted. Additional works related to dimension reduction and GPs have been proposed in (Fukutani et al., 2021; Rohmer et al., 2016; Perrin et al., 2021)

Many of the aforementioned works have in common that they consider scalar representations of the hydrodynamic forcing conditions (inputs) rather than their functional structures (e.g., time series). We particularly refer to (Azzimonti et al., 2019; Rohmer et al., 2018) for some examples where hydrodynamic functional drivers are parameterized by scalar representations before assessing the surrogate models. To date, the frameworks in (Betancourt et al., 2020; Kim et al., 2015) are the only frameworks that consider time-varying inputs in the domains of flood hazard assessment and storm surge prediction, respectively. Betancourt et al. (2020) further investigate dedicated kernels based on proper distances in function spaces aiming to correlate hydrodynamic functional drivers. Kim et al. (2015) introduce a time-dependent surrogate model of storm surge based on an artificial neural network.

Betancourt et al. (2020) have shown that considering hydrodynamic drivers as functions rather than scalars results in significant improvements in prediction as more precise physical information is encoded into kernels. Their work has focused on modeling, such as a scalar representation of the maximum cumulative overtopped and overflowed water volume. However, in practice, these global scalar indicators may be insufficient; and spatial information, e.g., the maximum inland water level, is often needed. Therefore, inspired by (Betancourt et al., 2020), our aim here is to establish a GP model that accounts for the effects of time-varying forcing conditions and provides spatially varying inland flood information. Our framework builds on the construction of a separable kernel that incorporates both functional and spatial correlations. The resulting process can be viewed as a multioutput GP (see, e.g. Alvarez et al., 2012) where spatial flood indicators (outputs) are driven by a set of hydrometeorological drivers (functional inputs). In our case, the outputs are correlated by a kernel that exploits the "similarity" between the functional inputs. This leads to a framework that can be easily connected to other GP-based approaches (Lázaro-Gredilla and Figueiras-Vidal, 2009). Furthermore, efficient implementations can be developed based on Kronecker-structured computations (see, e.g., Alvarez et al., 2012) or sparse-variational approximations (see, e.g., Van der Wilk et al., 2020). For the former case, we provide R codes based on the `kergp` package (Deville et al., 2019); and for the latter case, we adapt the multioutput GP models from the `GPflow` library (Matthews et al., 2017) to account for functional input data.

There are alternative approaches that treat spatial outputs as functional data. For instance, we refer to (Marrel et al., 2011) for a framework modeling the pollution produced by radioactive wastes, to (Chang and Guillas, 2019) for an approach capable of learning spatial patterns in climate experiments, and to (Perrin et al., 2021) for a surrogate model for the assessment of coastal flooding risks. These approaches first project the outputs into truncated basis representations (e.g., wavelets) to reduce the dimensionality. Then, prior distributions are placed at the level of the representation coefficients and not in the output space. Although their works scale well with the number of spatial points, they typically require a large number of learning simulations (e.g., over 500 events) to properly capture spatial patterns (see, e.g., Perrin et al., 2021). Here, we are restricted to highly constraining situations where fewer than 200 flood scenarios are available. This limitation arises from the use of costly-to-evaluate numerical simulators: a flood event requires almost three days of parallel computing (see, e.g., the model used by Idier et al., 2020b). We must also point out that (Marrel et al., 2011; Chang and Guillas, 2019; Perrin et al., 2021) do not account for functional inputs, a condition that is crucial for properly learning hydrometeorological forcing conditions, as our framework does (see the discussion in Section 2.1). To the best of our knowledge, our proposal is the first to consider both inputs and outputs as functions in the field of flood hazard assessment. Furthermore, unlike the aforementioned works, our framework allows us to focus predictions on spatial design points placed in key sectors identified as of uttermost importance regarding the vulnerability of the territory (see, e.g., Idier et al., 2013, 2020a, for a further discussion).

Map Data: Google, Maxar Technologies.

Figure 1: (left) The Gâvres municipality (red contour). (right) The town of Gâvres (*Grande Plage* and the cemetery).

The remaining sections are organized as follows. In Section 2, we describe the coastal flooding application that motivated the contributions in this paper. We briefly explain how to establish GP models for functional data, and we introduce the extension to spatial GPs with functional data where the covariance function is built via separable kernels. We also discuss the connection of the proposed framework with multioutput GP models. In Section 3, we assess the performance of the resulting GP model on various synthetic examples considering different situations depending on the data availability. We also apply our framework to a coastal flooding application. Finally, in Section 4, we summarize our results and outline potential future work.

2 Study Site, Data and Methods

In this section, we first describe the study site and the dataset used in the numerical experiments of Section 3.4. We then provide the background related to Gaussian process models with time-varying functional inputs. Finally, we introduce the extension to the case of spatial outputs.

2.1 Coastal flooding application: study site and dataset

In our study, we focus on the town of Gâvres (Figure 1, left) located along the French Atlantic coast in Brittany. The Gâvres municipality is a peninsula that is connected to the mainland by a 6 km long tombolo (Figure 1, right). The town has faced five significant coastal flood events since 1900 (Idier et al., 2020b). The latest memorable event occurred on 10 March 2008 (Johanna storm): a combination of a spring tide, a storm surge larger than 0.5 m, and energetic waves led to the flooding of approximately 120 houses (Idier et al., 2020b), some by approximately 1 m of water, along the street of the sports park (Le Roy et al., 2015; André et al., 2013). This marine submersion was induced mainly by waves overtopping the sea dike at the *Grande-Plage* beach and a bit of overflow close to the cemetery (Figure 1, right).

In such environments, estimating floods requires the use of advanced hydrodynamic numerical models able to account for overflow and overtopping processes with good precision. Such models emulate the hydrodynamics (water level and current) induced by hydrometeorological forcing conditions (e.g., the mean sea level, tide, atmospheric surge, and wave conditions). An example is the nonhydrostatic phase-resolving SWASH model (Zijlema et al., 2011). The SWASH model, nested with a spectral wave model that propagates the offshore wave conditions to the SWASH model's boundary, can successfully reproduce local past flood events (see Idier et al., 2020b). However, this modeling configuration is time consuming (6 h simulated in 3 days on 48 cores) and is therefore inapplicable for flood forecasting.

Thus, to support the development of fast-running surrogate models for flood forecasting, the BRGM and IMT[1] built a dataset $(X_o \to Y_o)$ based on numerical modeling. In the present work, we use this dataset. The hydrometeorological forcing conditions (X_o) are time series of the mean sea level (MSL [m]), tide (T [m]), atmospheric storm surge (S [m]), significant wave height (Hs [m]), wave peak period (Tp [s]), wave peak direction (Dp [°]), wind speed (U [m/s]) and wind direction (Du [°]) (see Idier et al., 2020b, for more details). These drivers, discretized with a 10 min time step over a 6 h window centered on high tide, are represented by a 37 observation long time series. The stored model results (Y_o) are the maximal inland water height $(H_{\max})$ at each grid point every 3 m (see Figure 2). This leads to

[1]BRGM: The French Geological Survey (acronym in French). IMT: Toulouse Mathematics Institute (acronym in French).

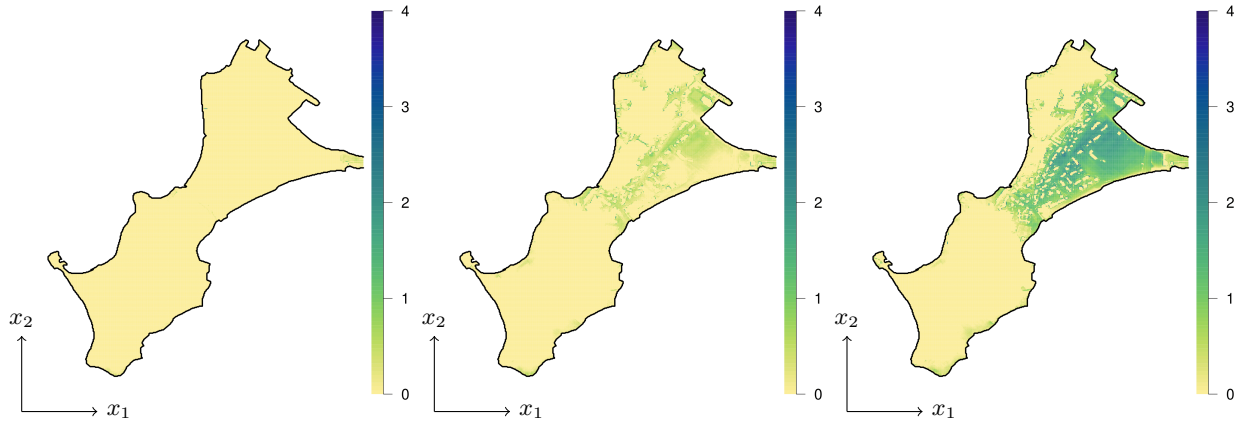


Figure 2: Three flood examples from the dataset in Section 2.1. The panels show the maximal inland water level $H_{\max}$ [m].

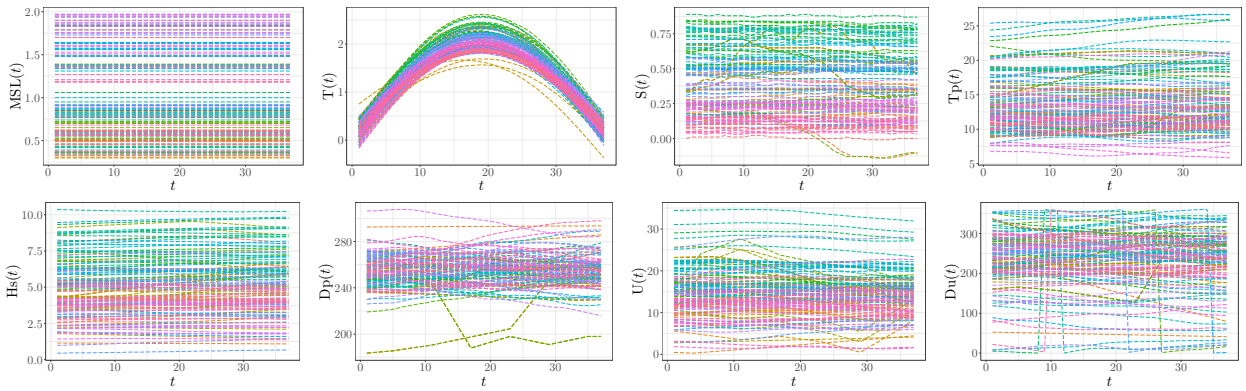


Figure 3: Hydrometeorological functional inputs from the dataset in Section 2.1. The panels show the 131 replicates of the mean sea level (MSL [m]), tide (T [m]), surge (S [m]), wave peak period (Tp [s]), wave height (Hs [m]), wave peak direction (Dp [°]), wind speed (U [m/s]) and wind direction (Du [°]).

spatial flood events containing approximately 64.6k inland observations. The dataset contains 131 scenarios of X_o, including 21 historical flood (9) and no flood (12) events (see Idier et al., 2020b); 16 scenarios simulated from small variations of the 9 historical flood events; and 94 additional scenarios with both zero, moderate and significant marine submersions. The latter (94) scenarios were built by applying a combination of methods to a hydrometeorological dataset covering the 1900-2016 period with a 10 min time step. Namely, multivariate extreme value analysis was applied to randomly generate the joint distribution of the maximum values of forcing conditions, a probabilistic classifier was applied to locate the time instant of these maximum values, and a multivariate Gaussian Monte Carlo-based sampling procedure was applied to generate the time series accordingly. We refer to (Idier et al., 2020b) for the dataset description and (Rohmer and Idier, 2012; Idier et al., 2020b) for the extreme value analysis. The 131 scenarios of the hydrometeorological functional inputs are shown in Figure 3.

2.2 Background on Gaussian processes with functional inputs

2.2.1 Gaussian processes

GP-based surrogate models have been widely used to replace costly-to-evaluate numerical simulators due to their well-founded and nonparametric framework for statistical learning (Rasmussen and Williams, 2005; Camps-Valls et al., 2016). GPs form a flexible prior over functions where regularity assumptions can be encoded into covariance functions also known as kernels (Genton, 2001; Paciorek and Schervish, 2004).

Kernel	Mathematical expression $k_{\boldsymbol{\theta}=(\sigma^2,\boldsymbol{\ell})}(\boldsymbol{\mathcal{R}})$
Squared exponential (SE)	$\sigma^2 \exp\left\{-\frac{1}{2}\|\boldsymbol{\mathcal{R}}\|_{\boldsymbol{\ell}}^2\right\}$
Matérn 5/2	$\sigma^2\left(1+\sqrt{5}\|\boldsymbol{\mathcal{R}}\|_{\boldsymbol{\ell}}+\frac{5}{3}\|\boldsymbol{\mathcal{R}}\|_{\boldsymbol{\ell}}^2\right)\exp\left\{-\sqrt{5}\|\boldsymbol{\mathcal{R}}\|_{\boldsymbol{\ell}}\right\}$
Matérn 3/2	$\sigma^2\left(1+\sqrt{3}\|\boldsymbol{\mathcal{R}}\|_{\boldsymbol{\ell}}\right)\exp\left\{-\sqrt{3}\|\boldsymbol{\mathcal{R}}\|_{\boldsymbol{\ell}}\right\}$
Exponential	$\sigma^2 \exp\left\{-\|\boldsymbol{\mathcal{R}}\|_{\boldsymbol{\ell}}\right\}$

Table 1: Examples of valid stationary kernel functions

Let $\{Y(\boldsymbol{\mathcal{F}}); \boldsymbol{\mathcal{F}} \in \mathcal{F}(\mathcal{T},\mathbb{R})^Q\}$, where $\mathcal{F}(\mathcal{T},\mathbb{R})$ is the set of functions from $\mathcal{T}$ to $\mathbb{R}$, be a stochastic process with functional inputs $\boldsymbol{\mathcal{F}} = (f_1,\ldots,f_Q)$. Consistent with Section 2.1, here, we consider time-varying inputs $f:\mathcal{T}\to\mathbb{R}$ with $\mathcal{T}\subseteq\mathbb{R}$. We must note that our developments can be extended to the multivariate case, i.e., $\mathcal{T}\subseteq\mathbb{R}^D$ with $D\geq 2$.

We say that Y is GP-distributed if any finite subset of random variables extracted from Y has a joint Gaussian distribution (Rasmussen and Williams, 2005). By focusing on centered GP priors,[2] then Y is completely defined by

$$Y \sim \mathcal{GP}(0,k), \tag{1}$$

where the kernel $k(\boldsymbol{\mathcal{F}},\boldsymbol{\mathcal{F}}')$ evaluates the correlation between $Y(\boldsymbol{\mathcal{F}})$ and $Y(\boldsymbol{\mathcal{F}}')$. For instance, $k(\boldsymbol{\mathcal{F}},\boldsymbol{\mathcal{F}}')=0$ if $Y(\boldsymbol{\mathcal{F}})$ and $Y(\boldsymbol{\mathcal{F}}')$ are uncorrelated and is nonzero otherwise.

One of the main benefits of GPs relies on the tractability of conditional distributions. Let us consider Y conditioned on an observation vector $\boldsymbol{y}_N=[y_1,\ldots,y_N]^\top$ evaluated at $(\boldsymbol{\mathcal{F}}_1,\ldots,\boldsymbol{\mathcal{F}}_N)$, where $\boldsymbol{\mathcal{F}}_i=(f_{i,1},\ldots,f_{i,Q})$ for $i=1,\ldots,N$. Then, the conditional distribution $Y|\{\mathbf{Y}_N=\boldsymbol{y}_N\}$ for the Gaussian vector $\mathbf{Y}_N=[Y(\boldsymbol{\mathcal{F}}_1),\ldots,Y(\boldsymbol{\mathcal{F}}_N)]^\top$ is also Gaussian with conditional mean and conditional covariance functions given by

$$\begin{aligned}\mu(\boldsymbol{\mathcal{F}}_*) &= \boldsymbol{k}^\top(\boldsymbol{\mathcal{F}}_*)\mathbf{K}^{-1}\boldsymbol{y}_N,\\ c(\boldsymbol{\mathcal{F}}_*,\boldsymbol{\mathcal{F}}_*') &= k(\boldsymbol{\mathcal{F}}_*,\boldsymbol{\mathcal{F}}_*') - \boldsymbol{k}^\top(\boldsymbol{\mathcal{F}}_*)\mathbf{K}^{-1}\boldsymbol{k}(\boldsymbol{\mathcal{F}}_*'),\end{aligned} \tag{2}$$

with covariance matrix $\mathbf{K}=(k(\boldsymbol{\mathcal{F}}_i,\boldsymbol{\mathcal{F}}_j))_{1\leq i,j\leq N}$ and cross-covariance vector $\boldsymbol{k}(\boldsymbol{\mathcal{F}})=[k(\boldsymbol{\mathcal{F}},\boldsymbol{\mathcal{F}}_1),\ldots,k(\boldsymbol{\mathcal{F}},\boldsymbol{\mathcal{F}}_N)]^\top$. The conditional mean $\widehat{Y}(\boldsymbol{\mathcal{F}}_*)=\mu(\boldsymbol{\mathcal{F}}_*)$ is usually used as a point estimate of $Y(\boldsymbol{\mathcal{F}}_*)$, and the conditional variance $v(\boldsymbol{\mathcal{F}}_*)=c(\boldsymbol{\mathcal{F}}_*,\boldsymbol{\mathcal{F}}_*)$ is the expected square error of this estimate.

2.2.2 Construction of stationary kernels for functional inputs

To establish proper kernels, we need to define the distance between functions. We consider the L^2-norm as it leads to simpler and closed-form expressions in Section 2.2.3:

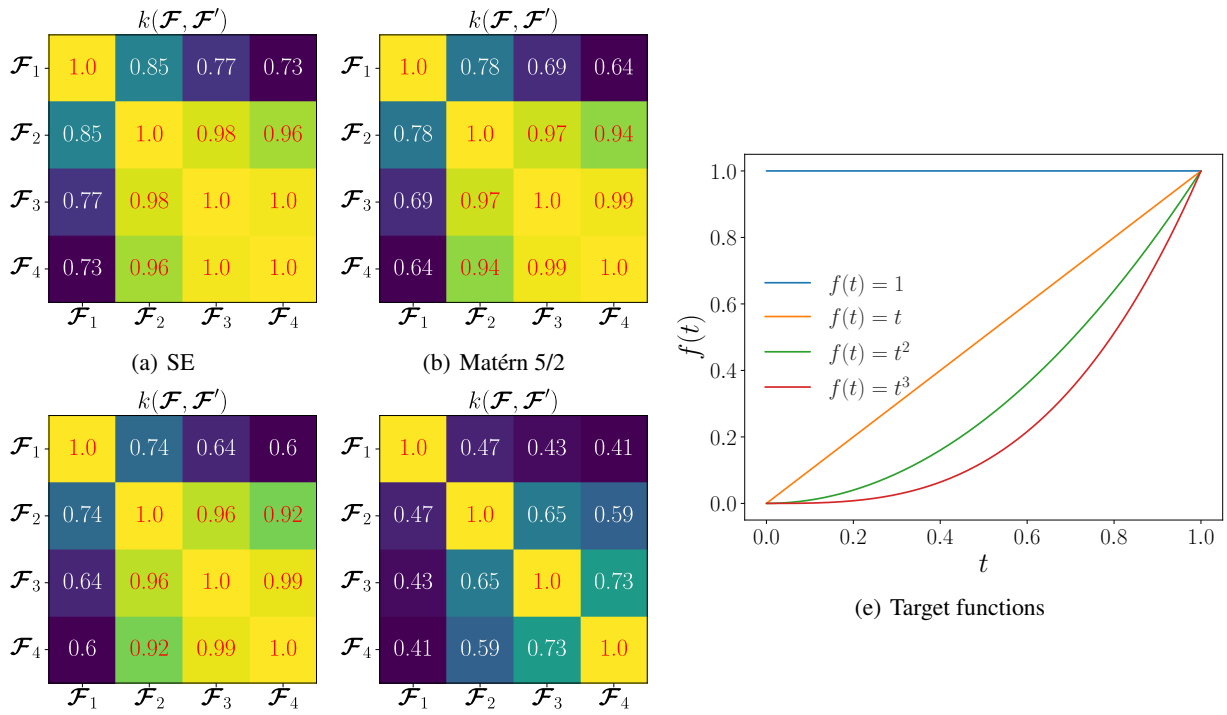
$$\|\boldsymbol{\mathcal{R}}\|_{\boldsymbol{\ell}}^2 = \|\boldsymbol{\mathcal{F}}-\boldsymbol{\mathcal{F}}'\|_{\boldsymbol{\ell}}^2 = \sum_{i=1}^{Q}\frac{\int_{\mathcal{T}}(f_i(t)-f_i'(t))^2dt}{\ell_i^2}, \tag{3}$$

where $\ell_i\in\mathbb{R}^+$ and $\int_{\mathcal{T}}(f_i(t)-f_i'(t))^2dt<\infty$, for $i=1,\ldots,Q$. The length-scale parameter ℓ_i can be viewed as a scale parameter for the ith functional input.

Examples of valid stationary kernels are shown in Table 1, and illustrations of the effect of those kernels are displayed in Figure 4. We consider $\boldsymbol{\mathcal{F}}_1=(f(t)=1)$, $\boldsymbol{\mathcal{F}}_2=(f(t)=t)$, $\boldsymbol{\mathcal{F}}_3=(f(t)=t^2)$ and $\boldsymbol{\mathcal{F}}_4=(f(t)=t^3)$ as inputs. We fix the variance parameter $\sigma^2=1$ and the length-scale parameter $\ell=1$. We can observe that the cross-covariance values $k(\boldsymbol{\mathcal{F}}_i,\boldsymbol{\mathcal{F}}_j)$, for $i,j=2,3,4$ with $i\neq j$, are greater than those involving $k(\boldsymbol{\mathcal{F}}_1,\boldsymbol{\mathcal{F}}_j)$. This results from the similarity between the functions $f(t)=t$, $f(t)=t^2$ and $f(t)=t^3$ on $\mathcal{T}=[0,1]$. Furthermore, from Figure 4(a) to 4(d), we note that GP models can be more or less sensitive to dissimilarities depending on the choice of the kernel. For example, while the squared exponential (SE) kernel provides a correlation between $\boldsymbol{\mathcal{F}}_2$ and $\boldsymbol{\mathcal{F}}_4$ equal to one (Figure 4(a)), the exponential kernel suggests that the similarity between those inputs is weaker and equal to 0.73 (Figure 4(d)).

The exact computation of (3) relies on its Q integrals. Depending on the complexity of f_i and f_i', such integrals can be intractable. Furthermore, in many situations, only a finite number of evaluations of f_i and f_i' are available, but their functional structures are actually unknown (e.g., for the case of discrete time series). For these reasons, functions are usually replaced by linear approximations where (3) has a closed-form solution since the integrals operate over well-defined basis functions.

[2] A GP Z with mean function μ and kernel k can be written in terms of a centered GP Y with same kernel: $Z(\boldsymbol{\mathcal{F}})=\mu(\boldsymbol{\mathcal{F}})+Y(\boldsymbol{\mathcal{F}})$.



(a) SE (b) Matérn 5/2 (c) Matérn 3/2 (d) Exponential (e) Target functions

Figure 4: Effect of the kernels in Table 1 with $\mathcal{F}_1 = (f(t) = 1)$, $\mathcal{F}_2 = (f(t) = t)$, $\mathcal{F}_3 = (f(t) = t^2)$ and $\mathcal{F}_4 = (f(t) = t^3)$ as functional inputs. The covariance parameters are set to $(\sigma^2 = 1, \ell = 1)$. The panels show the kernels (left) and the functions f (right).

2.2.3 Projection of functional inputs onto basis functions

Consider the projection of f onto a set of basis functions (Ramsay and Silverman, 2005):

$$f(t) \approx \sum_{j=1}^{p} \phi_j(t)\alpha_j := g(t). \tag{4}$$

Similarly, let g' be the linear approximation of f' with the same set of basis functions $\phi_1, \ldots, \phi_p$ and with the coefficients $\alpha'_1, \ldots, \alpha'_p$. Using (4), the integrals in (3) are then approximated by the form $\int_{\mathcal{T}} (g_i(t) - g'_i(t))^2 dt$. Note that in (4), for ease of notation, we intentionally dropped the subindex i in the definition of f_i.

Matricially, the integral in (4) is given by

$$\int_{\mathcal{T}} (g(t) - g'(t))^2 dt = \boldsymbol{\beta}^\top \boldsymbol{\Psi} \boldsymbol{\beta}, \tag{5}$$

where $\boldsymbol{\Psi} = \int_{\mathcal{T}} \boldsymbol{\phi}(t)\boldsymbol{\phi}^\top(t)dt$, $\boldsymbol{\phi}(t) = [\phi_1(t), \ldots, \phi_p(t)]^\top$, $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_p]^\top$ and $\beta_j = \alpha_j - \alpha'_j$. Note from (5) that the integral now operates over $\phi_1, \ldots, \phi_p$ whose functional structures are known.

For a wide range of basis families, such as in splines and PCA, the Gram matrix $\boldsymbol{\Psi}$ has an analytical form (see, e.g., Ramsay and Silverman, 2005; Shi and Choi, 2011). In terms of computational savings, since $\boldsymbol{\Psi}$ does not depend on f or f', it can be computed only once, stored and reused when building up the kernels in Table 1. The consideration of orthogonal or orthonormal families of basis functions (e.g., in PCA) results in significant simplifications. For the former family, $\boldsymbol{\Psi}$ is a diagonal matrix given by $\boldsymbol{\Psi} = \mathrm{diag}(\int_{\mathcal{T}} \boldsymbol{\phi}(t)\boldsymbol{\phi}^\top(t)dt)$. For the orthonormal case, $\boldsymbol{\Psi}$ is the identity.

In this paper, we focus on projections based on PCA approximations as they lead to two main computational benefits (see, e.g., Betancourt et al., 2020, for other implementations based on splines). First, $\boldsymbol{\Psi} = \mathbf{I}$ due to the orthonormality of the basis functions. Second, the most relevant information from functional inputs is encoded in a few principal components. As an example, to satisfy a total inertia of 99.9% for the 37 observation long discrete time series in Figure

3, PCA led to $\boldsymbol{p} = [1, 4, 3, 3, 2, 6, 4, 9]$ where p_i is the number of principal components for the ith functional input. Note that smaller values of p_i are assigned to less varied functional profiles. We refer to Appendix A for a further discussion on the construction of the PCA basis functions.

In the coastal flooding application, we consider discrete representations of the functional inputs. Therefore, we may be interested in exploiting classic measure distances such as the Euclidean or Mahalanobis distances; moreover, we may contemplate dedicated distance measures proposed for time series (see, e.g., Górecki and Piasecki, 2019; Mori et al., 2016). Note that those distances will consider 37-dimensional input spaces compared to the p-dimensional spaces, where $p \in \{1, 9\}$, obtained by the PCA projection. The increase in the computational costs here can be mild, but the PCA approximation will be better in other situations where the input space generated by the discrete time series is large. For example, if time series contain hundreds or thousands of discrete values, it is preferable to use the PCA projection to achieve cheaper and faster computations.

2.3 Extension to spatial Gaussian processes

We now consider that $\{Y(\boldsymbol{\mathcal{F}}, \boldsymbol{x}); \boldsymbol{\mathcal{F}} \in \mathcal{F}(\mathcal{T}, \mathbb{R})^Q, \boldsymbol{x} \in \mathbb{R}^2\}$ is a centered spatial GP with functional inputs $\boldsymbol{\mathcal{F}}$ and spatial coordinates $\boldsymbol{x} = (x_1, x_2)$. Hence, GP Y can be fully defined by constructing a valid kernel k that accounts for both spatial information and functional inputs, i.e., $k((\boldsymbol{\mathcal{F}}, \boldsymbol{x}), (\boldsymbol{\mathcal{F}}', \boldsymbol{x}'))$. Note that if the approximations $g_1, \ldots, g_Q$ are considered instead, then $k((\boldsymbol{\mathcal{F}}, \boldsymbol{x}), (\boldsymbol{\mathcal{F}}', \boldsymbol{x}'))$ must be rewritten for $k((\boldsymbol{\mathcal{G}}, \boldsymbol{x}), (\boldsymbol{\mathcal{G}}', \boldsymbol{x}'))$. For the sake of consistency with Sections 2.2.1 and 2.2.2, we continue the discussion with the notations of $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{\mathcal{F}}'$.

2.3.1 Construction of the covariance function via separable kernels

A natural extension of GPs accounting for mixed variables relies on separable kernels defined by the product of subkernels (see, e.g., Fricker et al., 2013; Roustant et al., 2020). In our case, the kernel can be written as

$$k((\boldsymbol{\mathcal{F}}, \boldsymbol{x}), (\boldsymbol{\mathcal{F}}', \boldsymbol{x}')) = k_f(\boldsymbol{\mathcal{F}}, \boldsymbol{\mathcal{F}}') k_x(\boldsymbol{x}, \boldsymbol{x}'), \tag{6}$$

where subkernels $k_x : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$ and $k_f : \mathcal{F}(\mathcal{T}, \mathbb{R})^Q \times \mathcal{F}(\mathcal{T}, \mathbb{R})^Q \to \mathbb{R}$.[3] Note now that the kernel in (6) is attenuated by the spatial correlation. Therefore, although having similar functional inputs $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{\mathcal{F}}'$, distant values of $\boldsymbol{x}$ and $\boldsymbol{x}'$ can result in small correlations. Since the process Y remains a GP, the conditional formulas in (2) hold for (6) with tuples $(\boldsymbol{\mathcal{F}}_i, \boldsymbol{x}_i)_{1 \le i \le N}$.

One of the main benefits of using (6) relies on the exploitation of Kronecker structures. In that case, we should consider tuples $(\boldsymbol{\mathcal{F}}_i, \boldsymbol{x}_j)_{1 \le i \le R, 1 \le j \le S}$, where R is the number of functional replicates (i.e. number of flood scenarios) and S is the number of spatial points per map. This leads to a total of $N = R \times S$ observations. Denote the covariance matrices as $\mathbf{K}_f = (k_f(\boldsymbol{\mathcal{F}}_i, \boldsymbol{\mathcal{F}}_j))_{1 \le i,j \le R}$ and $\mathbf{K}_x = (k_x(\boldsymbol{x}_i, \boldsymbol{x}_j))_{1 \le i,j \le S}$. Then, from (6), we have the Kronecker product $\mathbf{K} = \mathbf{K}_f \otimes \mathbf{K}_x$; and the Cholesky factorization of $\mathbf{K}$ given by $\mathbf{L} = \mathbf{L}_f \otimes \mathbf{L}_x$, where $\mathbf{L}_f$ and $\mathbf{L}_x$ are the (lower triangular) Cholesky matrices of $\mathbf{K}_f$ and $\mathbf{K}_x$, respectively. This results in less expensive procedures since both Cholesky and inverse operations are applied on smaller matrices, reducing the computational complexity to $\mathcal{O}(R^3) + \mathcal{O}(S^3)$ (compared to $\mathcal{O}(R^3 S^3)$ for standard implementations, Alvarez et al., 2012). For large datasets such as the one detailed in Section 2.1, computing either $\mathbf{K}$ or $\mathbf{L}$ (or their inverses) can easily run out of memory as either R or S increases. To mitigate this drawback, more efficient computations can be obtained by solving triangular-structured linear systems rather than directly computing those matrices (see, e.g., Bilionis et al., 2013). For easy self-reading, in Appendix B, we summarize the main Kronecker-based computations used in this paper.

2.3.2 Connection to other GP developments

Linear models of coregionalization (LMC). The process Y can be written as a multioutput process Z where the outputs are driven by a given set of functions, i.e., $Z_i(\boldsymbol{x}) := Y(\boldsymbol{\mathcal{F}}_i, \boldsymbol{x})$, for $i = 1, \cdots, R$. In that case, (6) yields:

$$k_{i,j}(\boldsymbol{x}, \boldsymbol{x}') = b_{i,j}\, k_x(\boldsymbol{x}, \boldsymbol{x}'), \tag{7}$$

where $b_{i,j} := k_f(\boldsymbol{\mathcal{F}}_i, \boldsymbol{\mathcal{F}}_j)$ for $i, j = 1, \ldots, R$. The kernel in (7) follows a similar structure to the one in LMC, more precisely, to the one in intrinsic coregionalization models (ICMs, Alvarez et al., 2012). Note that k_f involves only the estimation of Q length-scale parameters rather than the $R(R+1)/2$ coefficients ($R \gg Q$) corresponding to the upper triangular block of the symmetric coregionalization matrix $\mathbf{B} = (b_{i,j})_{1 \le i,j \le R}$. Another benefit of considering $\mathbf{K}_f$ is its positive definiteness condition since k_f is defined as a kernel. Since this condition is not necessarily satisfied by $\mathbf{B}$, ICMs may lead to numerical instabilities due to noninvertible matrices. In practice, $\widetilde{\mathbf{B}} = \mathbf{B} + \mathrm{diag}([\kappa_1, \ldots, \kappa_R])$ is considered to ensure the positive definiteness condition, but this implies the estimation of the R parameters $\kappa_1, \ldots, \kappa_R$.

[3] To avoid nonidentifiability of the variance parameters, k_f is considered a correlation function, i.e., the variance is $\sigma_f^2 = 1$.

Sparse approximations. Further simplifications are obtained by sparse approximations (Titsias, 2009; Hensman et al., 2013). Recall that $Z_i(\boldsymbol{x}) = Y(\boldsymbol{\mathcal{F}}_i, \boldsymbol{x})$ for $i = 1, \ldots, R$. The distribution of $(Z_1, \ldots, Z_R)$, conditioned to $(Z_i(\boldsymbol{x}_1) = z_{i,1}, \ldots, Z_i(\boldsymbol{x}_S) = z_{i,S})_{1 \leq i \leq R}$, can be approximated by a cheaper but tractable variational distribution conditioned to $(Z_i(\boldsymbol{u}_1) = \widetilde{z}_{i,1}, \ldots, Z_i(\boldsymbol{u}_M) = \widetilde{z}_{i,M})_{1 \leq i \leq R}$, with $M \ll S$. Then, using a low rank approximation of k_s, the complexity $\mathcal{O}(R^3) + \mathcal{O}(S^3)$ is reduced to $\mathcal{O}(R^3) + \mathcal{O}(SM^2)$. The inducing variables $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_M$ are estimated via variational inference (see, e.g., Hensman et al., 2013), and M is fixed seeking a trade-off between the computational complexity and approximation quality. While a large value of M leads to more precise but expensive models, a small value results in faster but less accurate approximations.

Variational inference (VI). As our functional framework preserves the structure of ICMs, it can be easily plugged into other types of multioutput GPs (MoGPs) based on VI (see, e.g., Hensman et al., 2013; Moreno-Muñoz et al., 2018; Van der Wilk et al., 2020). As an example, our functional framework can be fitted via stochastic VI (SVI), which scales well for large values of S (Hensman et al., 2013). Considering a separable Kronecker-based kernel and applying SVI (under sparse assumptions) only to the spatial kernel, the complexity of the resulting GP decreases to $\mathcal{O}(R^3) + \mathcal{O}(BM^2)$ with batch size $B \ll S$. The value of B is manually fixed depending on the available storage capacity. More efficient variational implementations are obtained by using interdomain approximations (see Lázaro-Gredilla and Figueiras-Vidal, 2009; Van der Wilk et al., 2020). As another example, heterogeneous MoGPs can also be established by considering that each output has its own likelihood function (e.g. a Gaussian, a Bernoulli or a Poisson likelihood, Moreno-Muñoz et al., 2018).

Since our MoGP implementations are based on the `GPflow` Python library, they enjoy a great variety of dedicated VI developments (see the documentation in Matthews et al., 2017), including those for the SVI and heterogeneous MoGPs. Although the variational features in (Moreno-Muñoz et al., 2018) are not exploited in this paper, they are available for the scientific community for further research (see Section 3.1 for further details).

3 Results and Discussions

3.1 Python and R implementations

The `GPflow` Python library relies on variational inference to meet the twin challenges of nonconjugacy and scale (Matthews et al., 2017). To the best of our knowledge, `GPflow` does not support Kronecker-based composite kernels in its latest release. Therefore, implementing Kronecker-structured objects requires significant modifications at the root level. This motivates developments based on the `kergp` R package (Deville et al., 2019). The `kergp` package is not equipped with sparse-variational approximations as `GPflow` is. Nevertheless, the `kergp` package is sufficiently flexible to account for both functional data and Kronecker-structured composite kernels. Both Python and R implementations are available on GitHub: `https://github.com/anfelopera/spatfGPs`.

3.2 Performance indicators

We consider the three following performance indicators for the predictions: the root mean square error (RMSE), the $\mathcal{Q}^2$ value and the coverage accuracy (CA). The first two assess the quality of the predictive mean, and the CA evaluates the quality of the predictive variance. The $\mathcal{Q}^2$ criterion is given by

$$\mathcal{Q}^2 = 1 - \frac{\sum_{i=1}^{N_{\text{test}}} (y_i - \widehat{y}_i)^2}{\sum_{i=1}^{N_{\text{test}}} (y_i - \overline{y})^2}, \tag{8}$$

where $\widehat{y}_1, \ldots, \widehat{y}_{N_{\text{test}}}$ and $\overline{y}$ are the predictions and the average of test data $y_1, \ldots, y_{N_{\text{test}}}$, respectively. For noise-free observations, $\mathcal{Q}^2$ is equal to one if predictions are exactly equal to the test data, zero if they are equal to $\overline{y}$, and negative if they perform worse than $\overline{y}$.

The CA of the c-standard deviation confidence intervals, denoted as $\text{CA}_{\pm c\sigma}$, indicates the proportion of the test data that is contained in the confidence intervals:

$$\text{CA}_{\pm c\sigma} = \sum_{i=1}^{N_{\text{test}}} \mathbb{1}_{y_i \in [\widehat{y}_i - c\sigma, \widehat{y}_i + c\sigma]}, \tag{9}$$

where σ^2 is the predictive variance provided by the GP model, and $\mathbb{1}_{y_i \in [\widehat{y}_i - c\sigma, \widehat{y}_i + c\sigma]}$ is equal to one if $y_i \in [\widehat{y}_i - c\sigma, \widehat{y}_i + c\sigma]$ and zero otherwise.

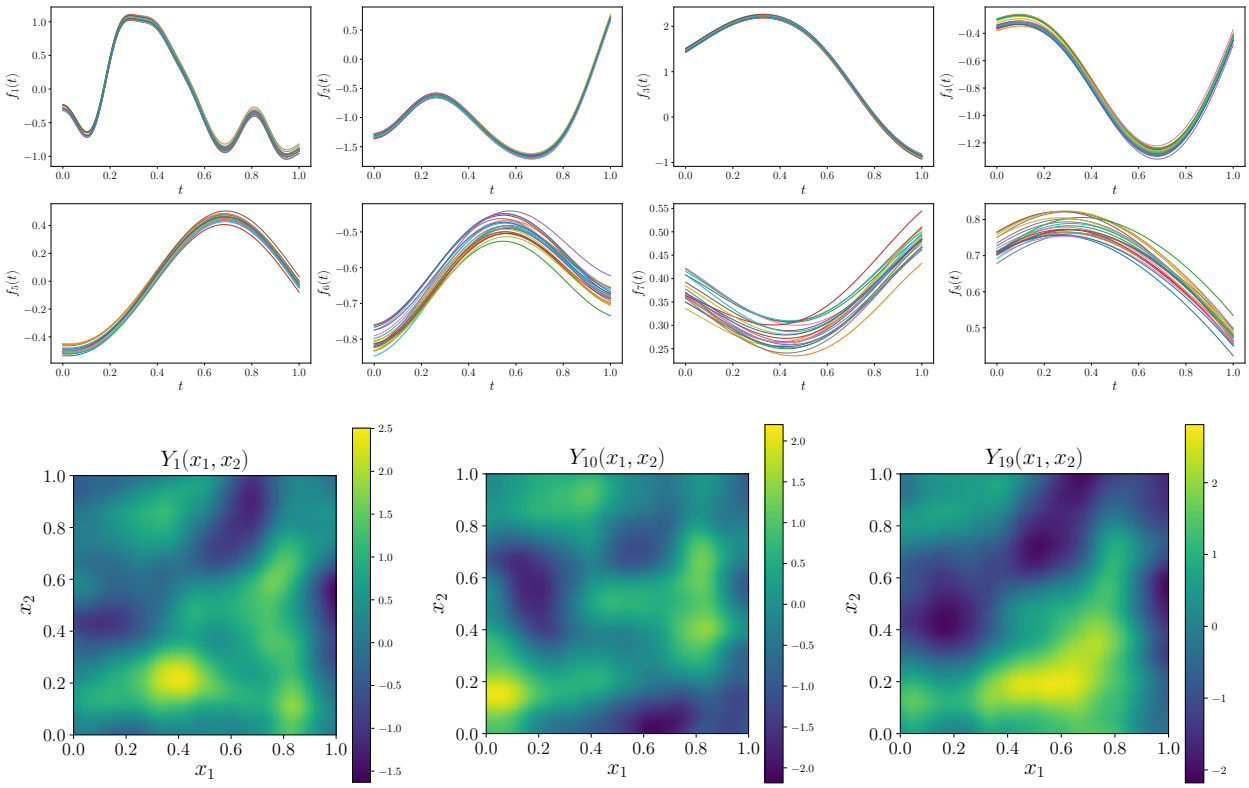


Figure 5: (top) Sampled replicates for the (8) inputs and (bottom) examples of sampled spatial events (20) used in the illustration of Section 3.3.1.

3.3 Numerical illustrations on synthetic examples

We test the proposed GP framework under different situations depending on the data availability. First, we consider the case where different design points per map are considered. To simplify the multioutput learning task, strong correlations between spatial events and functional inputs are assumed. Second, highly variable maps are predicted considering Kronecker-structured design points. Finally, we apply the model for predicting unobserved events given a set of learning events (forecasting task).

3.3.1 Multioutput illustration

Synthetic dataset. We consider a coupled system consisting of $R = 20$ outputs and $Q = 8$ inputs. To emulate functional patterns, the latter functions are sampled from the predefined GPs given by $f_i \sim \mathcal{GP}(\mu_i, k_o)$ for $i = 1, \ldots, Q$, with mean function μ_i and Matérn 5/2 kernel k_o. We fix the variance $\sigma_o^2 = 2.5 \times 10^{-3}$ and the length-scale $\ell_o = 0.8$ seeking small variations between f_i and μ_i. The mean functions $\mu_1, \ldots, \mu_Q$ are also GP realizations, i.e., $\mu_i \sim \mathcal{GP}(0, k_i)$, with Matérn 5/2 kernels, variances $\sigma_{\mu_i}^2 = \frac{1}{2}$ and length-scales $\ell_{\mu_i} = \frac{i}{10}$. Note that f_i becomes less variable as ℓ_{μ_i} increases (Figure 5). Then, we sample 100×100 maps using a 2D Matérn 5/2 kernel with variance $\sigma_x^2 = 1$ and length-scales $\ell_{x,1} = \ell_{x,2} = 0.2$. To correlate the inputs, a Matérn 5/2 kernel is used with length-scales $\ell_{f,i} = 2$ for $i = 1, \cdots, 8$. For convenience, strong correlations are considered to determine sampled maps that resemble each other (Figure 5).

Exact multioutput learning. We now compare the performance of the proposed functional MoGP (denoted as fMoGP) to the performance of the standard MoGP via LCM. For the MoGP, we use the developments available in `GPflow` (Matthews et al., 2017). Both models consider Matérn 5/2 kernels with the same initial spatial parameterization: $\sigma_{x,o}^2 = 1$ and $\ell_{x,1,o} = \ell_{x,2,o} = 0.5$. For the MoGP, as suggested by Matthews et al. (2017), we randomly initialize the coefficients of $\mathbf{B}$. For the fMoGP, a Matérn 5/2 kernel is used with initial length-scales $\ell_{f,i,o} = 0.5$ for $i = 1, \ldots, 8$. Then, the hyperparameters and covariance parameters of both models are estimated via maximum likelihood (ML) using $(\sigma_{x,o}^2, \ell_{f,1,o}, \ldots, \ell_{f,8,o})$ as the initial values of a gradient-based optimization. The gradients are computed using the automatic differentiation tool from `GPflow`.

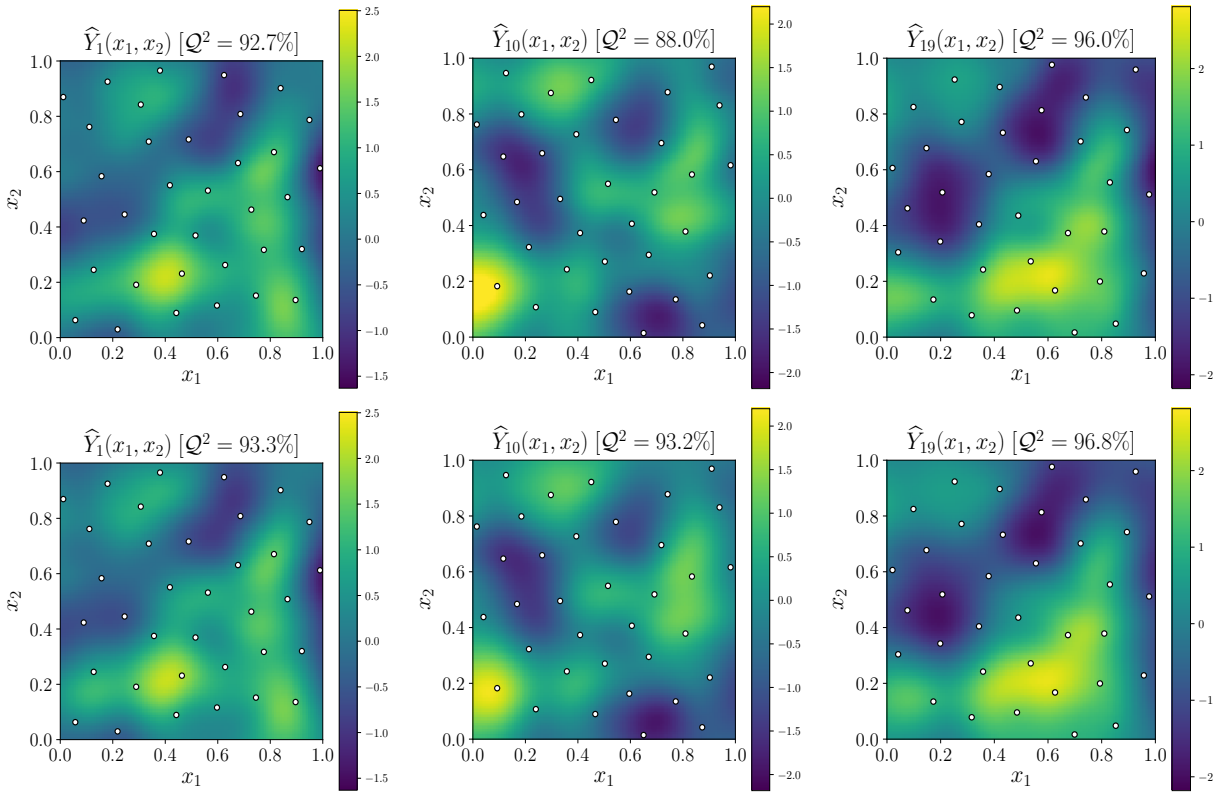


Figure 6: Predictions for the spatial events in Figure 5. The results are shown using either the standard MoGP via LMC (top) or the proposed functional MoGP framework (bottom). The white dots represent the spatial design points. The resulting $\mathcal{Q}^2$ value per map is also shown on top of each panel.

In the training step, we consider different maximin Latin hypercube designs (LHDs) with $S = 35$ spatial points per map, leading to a total of 700 training points.[4] We use the enhanced stochastic evolutionary (ESE)-based LHD implementation in the `SMT` Python library (Bouhlel et al., 2019; Jin et al., 2005). Note that since tensorized designs are not considered, Kronecker-structured models cannot be exploited.

Results. Figure 6 shows the predictions provided by both MoGP and fMoGP. We can observe that both models capture the spatial dynamics of the maps in Figure 5, leading to $\mathcal{Q}^2$ values of approximately 0.943 and 0.960 (averaged values over the 20 maps), respectively. Here, the $\mathcal{Q}^2$ criterion is computed over an equispaced 100×100 grid fixed for all the maps $Y_1, \ldots, Y_{20}$. After repeating this experiment using ten different LHDs, the MoGP and fMoGP led to $\mathcal{Q}^2 = 93.2 \pm 1.4$ % (mean $\pm$ standard deviation) and $\mathcal{Q}^2 = 95.9 \pm 0.6$ %, respectively. This raises the conclusion that accounting for the functional structure in the coregionalization matrix results in prediction improvements.

Although both matrices $\mathbf{B}$ and $\mathbf{K}_f = (k_f(\boldsymbol{\mathcal{F}}_i, \boldsymbol{\mathcal{F}}_j))_{1 \leq i,j \leq R}$ exhibit the strongest correlations in the diagonal (Figure 7), $\mathbf{B}$ led to extremely high values. This affects the estimation of the variance, leading to $\widehat{\sigma}^2_{x,\text{MoGP}} = 5.8 \times 10^{-4}$ compared to $\sigma^2_x = 1$ (true variance) and $\widehat{\sigma}^2_{x,\text{fMoGP}} = 0.96$. For the length-scales of k_x, models led to $\hat{\boldsymbol{\ell}}_{x,\text{MoGP}} = (0.20, 0.22)$ and $\hat{\boldsymbol{\ell}}_{x,\text{fMoGP}} = (0.20, 0.21)$ compared to the true results fixed to $\boldsymbol{\ell}_x = (0.2, 0.2)$. For the length-scales of k_f, the fMoGP estimated $\hat{\boldsymbol{\ell}}_f = (2.57, 1.84, 1.83, 1.66, 1.74, 3.55, 1.46, 1.53)$ compared to the true results fixed to 2.

3.3.2 Sparse-variational illustration

Synthetic dataset. The synthetic dataset is generated by following the sampling scheme used in Section 3.3.1 but with some covariance parameters slightly changed to find more variable spatial events. We fix $\sigma^2_o = 4 \times 10^{-2}$ and

[4] The budget has been fixed considering the computational capacity of an Intel(R) Core(TM) i5-4300M CPU @ 2.60GHz and 8 Gb RAM.

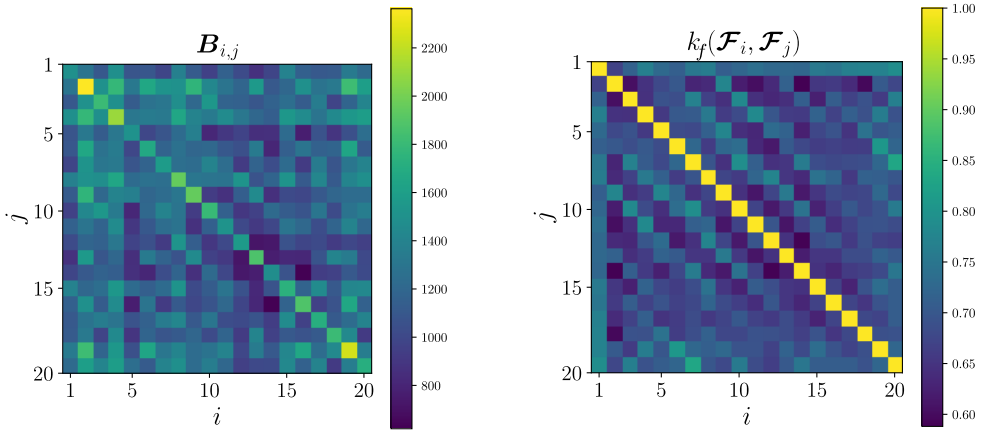


Figure 7: Coregionalization matrices for the MoGP (left) and fMoGP (right) models used in Section 3.3.1.

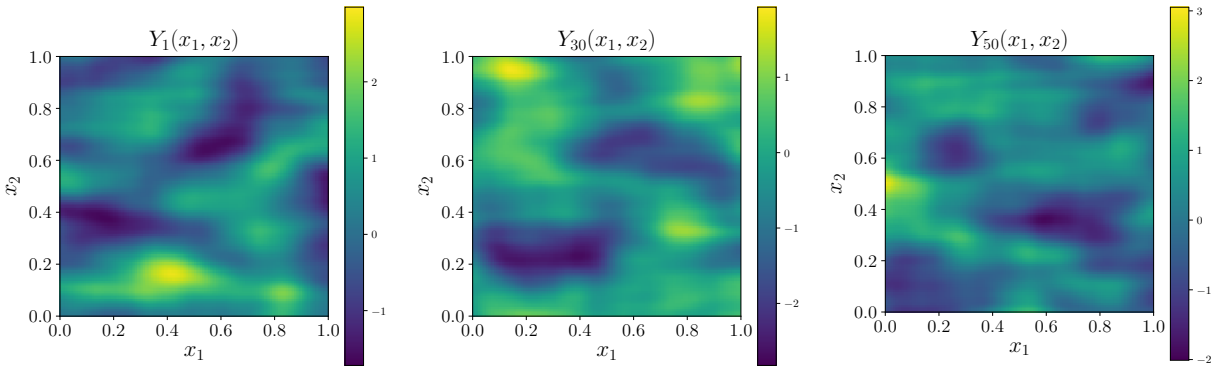


Figure 8: Examples of the sampled maps used in the illustration of Section 3.3.2.

$\ell_{x,1,o} = \ell_{x,2,o} = 0.1$. We then sample 50 maps using a 100×100 grid. Examples of the generated spatial events are shown in Figure 8.

Approximate multioutput learning. We have adapted the sparse-variational MoGP (sv-MoGP) framework in (Van der Wilk et al., 2020) to account for functional inputs. We denote the resulting model as sv-fMoGP. The efficiency of sv-MoGP relies on considering sparse approximations together with dedicated SVI schemes and tensor-structured data.

For the spatial design points, we use a maximin LHD of $S = 500$ points. This results in $N = 25 \times 10^3$ training data, a significantly larger amount of data compared to the nontensorized illustration proposed in Section 3.3.1. The 50 spatial events are assumed to share the same inducing variables.[5] Those variables are initialized as a maximin LHD seeking to cover the spatial space. To assess the influence of the inducing variables, we test models where the value of M is gradually increased, i.e., $M = 20, 50, 100, 200$. In order to avoid running out of memory,[6] we apply the SVI with a minibatch size of $B = 200$. Both the covariance and variational parameters are estimated via SVI based on gradient-based optimization using automatic differentiation (see Hensman et al., 2013; Van der Wilk et al., 2020).

Results. We tested both sv-MoGP and sv-fMoGP using ten different designs with $S = 500$ spatial points. The $\mathcal{Q}^2$ results (mean $\pm$ standard deviation) are shown in Table 2. The $\mathcal{Q}^2$ criterion was computed considering the spatial locations that were not used for training the models for all 50 maps. From Table 2, we observe that sv-fMoGP outperformed sv-MoGP, leading to absolute $\mathcal{Q}^2$ improvements greater than 12%. We can also note that sv-fMoGP resulted in accurate predictions for $M \geq 100$.

Although the predictability of both models improves as M increases, fitting them to training data becomes costly. While for $M = 100$ the CPU times for 4×10^3 gradient evaluations of the SVI were approximately 0.32 h and 0.74 h

[5]This assumption can be relaxed to address different sets of inducing variables per map but at the cost of complex inference.

[6]The experiments here have been executed on a single core of an Intel(R) Core(TM) i5-4300M CPU @ 2.60 GHz and 8 Gb RAM.

Table 2: Multioutput learning performance of sv-MoGP and sv-fMoGP. The $\mathcal{Q}^2$ results (mean $\pm$ standard deviation) are computed considering 10 replicates of the experiment in Section 3.3.2.

	$\mathcal{Q}^2$ [%] (mean $\pm$ standard deviation)			
Model	Number of Inducing Variables			
	$M = 20$	$M = 50$	$M = 100$	$M = 200$
sv-MoGP	53.6 ± 3.2	75.1 ± 2.1	81.4 ± 1.5	84.0 ± 0.8
sv-fMoGP	$\mathbf{69.9 \pm 1.7}$	$\mathbf{87.4 \pm 0.9}$	$\mathbf{93.6 \pm 1.2}$	$\mathbf{96.6 \pm 0.1}$

(for sv-MoGP and sv-fMoGP, respectively), for $M = 200$, they were approximately 0.96 h and 1.85 h, respectively. sv-fMoGP was much slower since the gradients with respect to the length-scales $\ell_{f,1}, \ldots, \ell_{f,8}$ were more expensive to evaluate.

3.3.3 Inference of unobserved outputs

Synthetic dataset. We consider a coupled system consisting of $R = 1001$ outputs and $Q = 8$ functional inputs. We sample the latter functions as proposed in Section 3.3.1 but consider centered processes: $f_i \sim \mathcal{GP}(0, k_i)$ with Matérn 5/2 kernels, $\sigma_i^2 = \frac{1}{2}$ and $\ell_i = \frac{i}{10}$ for $i = 1, \ldots, q$. We generate the corresponding 1001 maps using an equispaced grid of 10×10 spatial locations. This leads to $S = 100$ design points per map. We use the same kernel parameterization proposed in Sections 3.3.1 and 3.3.2.

Inference of new maps. As discussed in Section 2.1, our goal is to predict the consequence (i.e., the map of maximum water level $H_{\max}$) of unobserved storm events. This is achieved by correlating the hydrometeorological drivers (functional inputs). The length-scales $\ell_1, \ldots, \ell_Q$ of kernel k_f can be estimated using data from the observed events. Therefore, since the structure of k_f is completely defined after fitting the GP, our model can be applied for forecasting purposes. The framework in (Van der Wilk et al., 2020) requires the estimation of the mean and covariance functions of the variational distribution, which can only be estimated for observed outputs. This makes both sv-MoGP and sv-fMoGP inapplicable to forecasting tasks. However, we can still exploit Kronecker-based composite kernels using `kergp`.[7]

In this example, we are interested in the inference of the 1001st map using data from the first 1000 maps. The performance of the model is assessed in terms of the number of learning maps R. Here, the covariance parameters are estimated via ML using the derivative-free constrained optimizer by linear approximations (COBYLA) implemented in the `NLOpt` package (Johnson, 2020). We have also tested gradient-based optimizers, but they led to more expensive procedures.

Results. Figure 9 shows predictions using one standard deviation confidence intervals. We can observe that the predictive mean becomes closer to the test data as R increases, with a $\mathcal{Q}^2$ improvement of 24.1% between the results using 5 and 1000 learning events. Moreover, the model only needed $R = 200$ learning events to achieve accurate predictive performance of $\mathcal{Q}^2 \sim 90\%$. In terms of the uncertainties, the predictive intervals cover the test data and decrease as R increases.

3.4 Coastal flooding application

We now apply our framework to the coastal flooding application described in Section 2.1. As we focus on forecasting purposes, we use our R implementations based on `kergp` (Deville et al., 2019).

3.4.1 Numerical settings

Data preprocessing. Some of the hydrometeorological forcing conditions, such as the wave peak direction Dp and the wind direction Du, are defined using nautical conventions (i.e., with north considered the reference). Then, as Du represents an angle, a slight change in winds coming from the north may lead to high angle variations and therefore to complex PCA representations. To avoid this drawback, in our experiments, we replace the tuples $(\mathrm{Hs}, \mathrm{Dp})$ and $(\mathrm{U}, \mathrm{Du})$ with the Cartesian tuples $(\mathrm{Hs}_x, \mathrm{Hs}_y)$ and $(\mathrm{U}_x, \mathrm{U}_y)$ given by $\mathrm{Hs}_x = \mathrm{Hs} \cdot \sin(\mathrm{Dp})$, $\mathrm{Hs}_y = \mathrm{Hs} \cdot \cos(\mathrm{Dp})$, $\mathrm{U}_x = \mathrm{U} \cdot \sin(\mathrm{Du})$ and $\mathrm{U}_y = \mathrm{U} \cdot \cos(\mathrm{Du})$. This results in a set of hydrometeorological functional inputs consisting of (MSL, T, S, Tp, Hs_x, Hs_y, U_x, U_y).

[7] Experiments here are executed on a single core of an HP cluster with an Intel biprocessor, 32×2.2 GHz cores, 64 GB RAM.

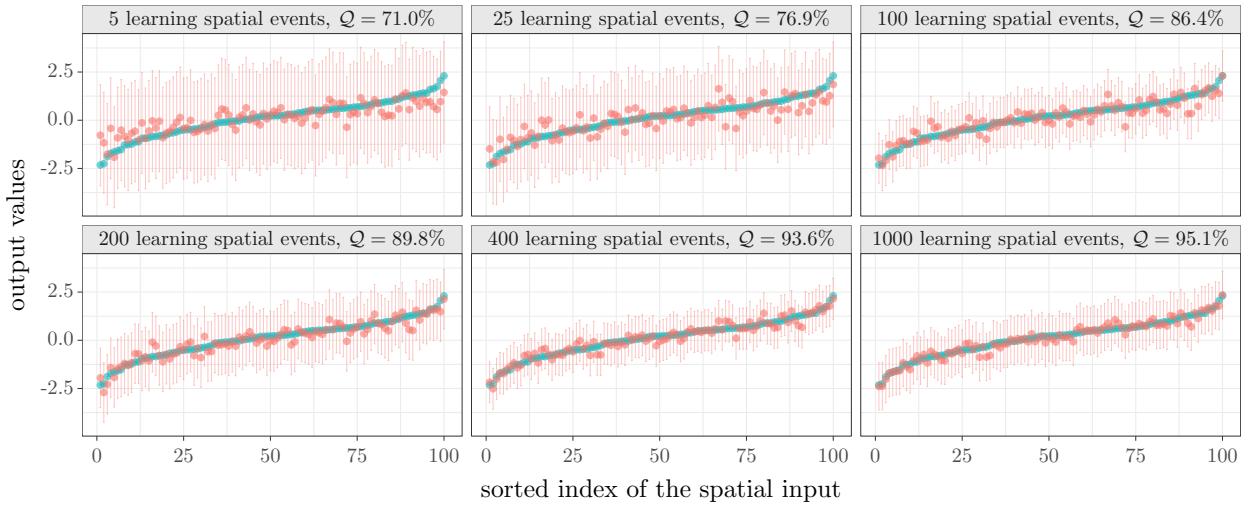


Figure 9: Predictions from Section 3.3.3 considering one standard deviation confidence intervals. Each panel shows the following: the sampled test data (blue dots), the predictive mean (red dots) and the confidence intervals (red whiskers). Outputs are sorted by increasing order of the sampled test data.

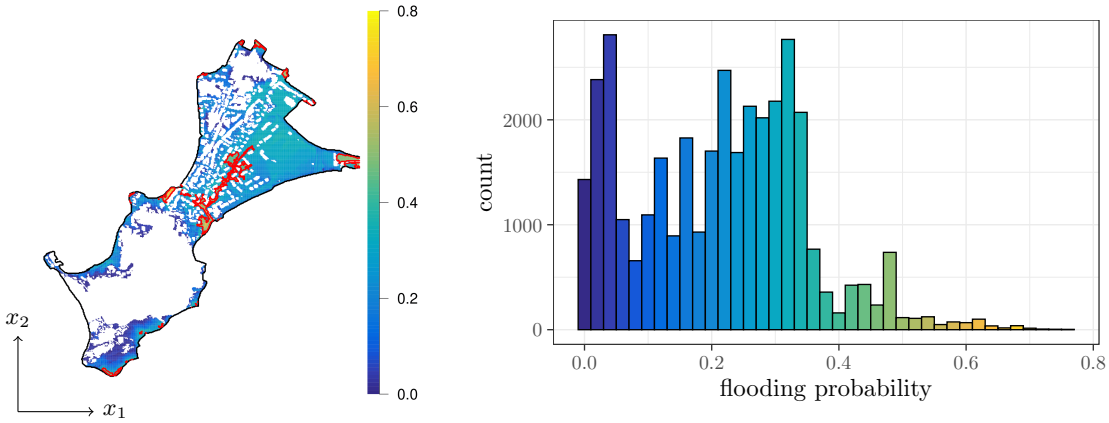


Figure 10: (left) Empirical flooding probability (EFP) over the 131 map replicates. (right) Histogram of the EFP. Both panels only consider nonzero values of the EFP. For the EFP, the contour line at 0.4 is shown in red.

Design of experiments (DoE). The complexity of our model increases with the number of spatial design points S. Therefore, we focus the computational budget on a subset of spatial locations. First, we consider locations where the empirical flooding probability (EFP) over the 131 flood scenarios is nonzero. This results in a total of 34×10^3 (approximately) possible candidates where EFP $\in (0, 0.8]$ (Figure 10, left). Second, we conveniently divide data into two classes. The DoE for the first class targets the neighborhood of the road network. The DoE of the second domain is space-filling over the entire domain. After analysis of the flood maps using expert knowledge, we noticed that spatial locations associated with the main road network led to EFP $\in [0.4, 0.8]$. Thus, we define the second class with locations where EFP $\in (0, 0.4)$. For each class, a k-means clustering scheme is applied (see, e.g., Hartigan and Wong, 1979), where the closest point to each cluster will be part of the DoE. The clustering scheme uses the spatial coordinates and the EFP as inputs, i.e., $\boldsymbol{x}_{\text{clustering}} = (x_1, x_2, \text{EFP})$. The influence of the EFP contributes to grouping spatially close points into different clusters if they exhibit different flooding probabilities. Note that the number of clusters κ_1, κ_2 are hyperparameters to be defined depending on how many spatial design points we consider per class. While κ_1 defines the desired number of design points with significant EFP values in the resulting DoE, κ_2 will control the number of design points used for space filling. We also consider three locations of interest from the neighboring district: the town hall, the gym and the lowest point of the sports field. They will be represented by a square, a circle and a triangle, respectively.

We must note that alternative approaches for the construction of DoEs, such as the PCA-based approach proposed by Zhang and Edgar (2008), could be considered. However, our aim here is not to provide optimal DoEs for further

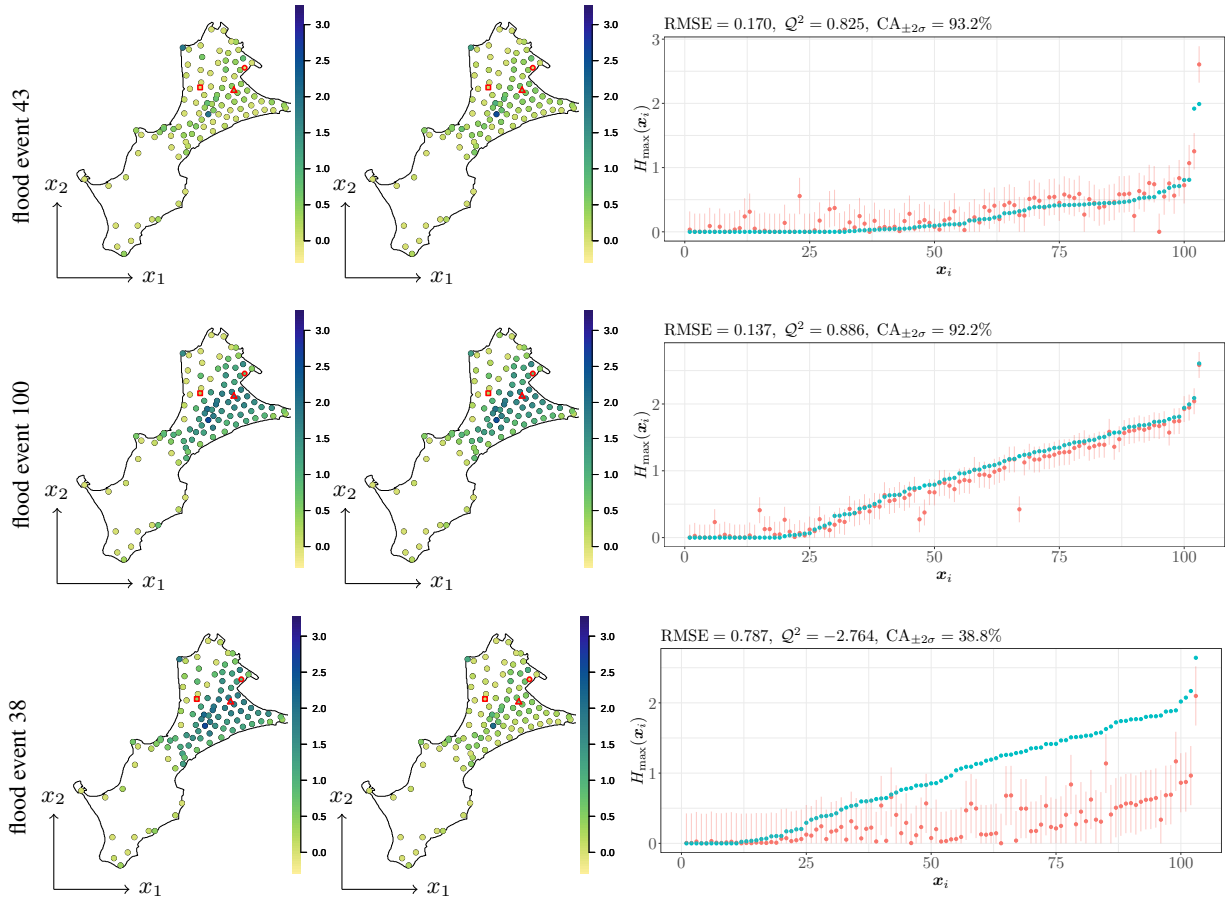


Figure 11: $H_{\max}$ predictions for the experiment in Section 3.4.2: (left) ground truth, (middle) predictions, and (right) ground truth (blue) vs. the predictive mean ($\pm$ two-standard deviation confidence intervals, red). For the flood event profiles, the town hall, gym and sports field are represented by a red square, a red circle and a red triangle, respectively. For the right side panels, the $H_{\max}$ values are sorted in increasing order of the true observations. The RMSE, $\mathcal{Q}^2$ and $CA_{\pm 2\sigma}$ values are shown on top of the right side panels.

implementations but to provide DoEs that contain spatial locations where floods have a strong impact while promoting space filling.

3.4.2 Leave-one-out (LOO) test

To validate our framework, we first test it on an LOO experiment where each scenario from the dataset is predicted using data from the other scenarios. This will give us an idea about the capability of the model for forecasting flood events. To train the 131 models, the same spatial design points are fixed for each scenario in order to apply Kronecker-based computations. For instance, we define a DoE with $S = 103$ locations (including the town hall, the gym and the sports field) using the k-means-based methodology discussed in Section 3.4.1. This leads to $N = 13390$ spatial points that are then used for the covariance parameter estimation. For the selection of the types of kernels used for correlating the functional inputs and spatial locations, different combinations of kernels in Table 1 have been tested. After running the corresponding LOO tests, the use of Matérn 5/2 kernels commonly outperformed any other combination. Therefore, the results here and in further experiments will consider functional and spatial Matérn 5/2 kernels.

Examples of the LOO predictions are shown in Figure 11. Since negative predictions do not have physical meaning in our application, we set them to zero for further analysis. From scenarios 43 and 100, our framework properly infers their flood levels, leading to accurate RMSE and $\mathcal{Q}^2$ values.[8] Note that for both scenarios, $CA_{\pm 2\sigma}$ covers more than 92% of the test data.

[8]The $\mathcal{Q}^2$ in (8) is invalid when the variance of the test data (the flood event that is predicted) is equal to zero. In our application, since some of the scenarios are not flooded, we redefine the $\mathcal{Q}^2$ criterion by normalizing the MSE using the variance of the 131 scenarios instead but considering only spatial locations where the EFP is nonzero.

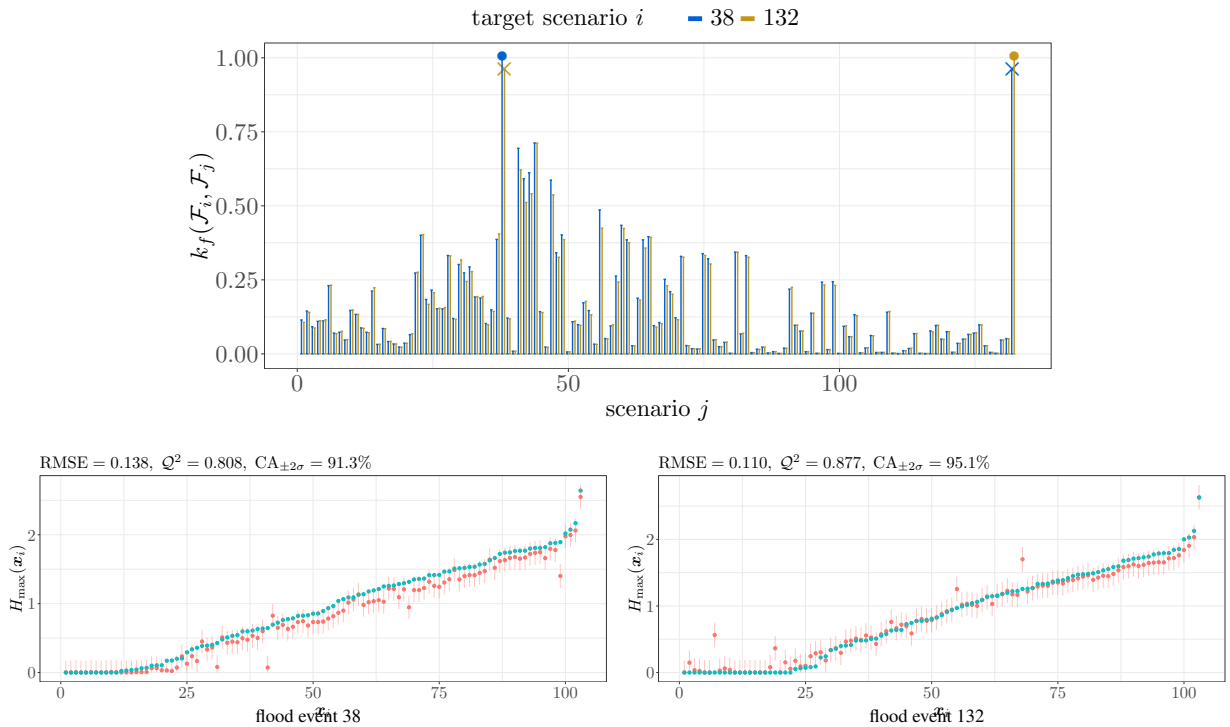


Figure 12: (top) Correlations between the hydrometeorological functional conditions of the predicted scenario and those of the learning set. (bottom) The $H_{\max}$ values (sorted in increasing order of the true observations) are shown after adding scenario 132. The panel description is the same as that of Figure 11.

Considering the predictions for the 131 LOO tests, they resulted in $\mathcal{Q}^2$ and $CA_{\pm 2\sigma}$ median values of approximately 95.8% and 99%, respectively. However, for some scenarios, our framework led to misprediction, as observed for the 38th scenario (see Figure 11). Indeed, in the coastal flooding dataset, there are scenarios that strongly differ from each other or that are unique in their type, and misprediction arises when one of those scenarios is analyzed and if there are no similar flood events available in the learning dataset. As we show in Figure 12, this drawback can be mitigated by incorporating additional and similar scenarios in the learning set of flood events. There, a new flood event, denoted as scenario 132, was added to the LOO test. The hydrometeorological conditions for this scenario are generated by slightly modifying the (Tp, Hs_x, Hs_y, U_x, and U_y) from the 38th scenario but preserving the same profiles for (MSL, T, and S). From Figure 12, the correlations between the hydrometeorological functional inputs of scenarios 38 and 132 are stronger than if we compare their inputs with those of other scenarios. Note also that predictions for the 38th scenario were significantly improved by exploiting data from the 132nd scenario, and vice versa.

3.4.3 Influence of the number of learning scenarios

As stated in Section 3.4.2, the performance of the model depends on the availability and diversity of learning flood scenarios. In this experiment, we stress the impact of enriching the set of learning scenarios for forecasting purposes. We focus on the prediction of (21) historical flood and no-flood events and (16) slightly reinforced historical flood events, i.e., the first 37 scenarios of the dataset. Those scenarios are predicted using data from the remaining 94 simulated scenarios. Since the latter set of scenarios was generated to cover a wide range of hydrometeorological conditions (see Section 2.1), accurate predictions are also expected when considering only a few learning scenarios.

We train models considering different subsets of the 94 learning scenarios. The selection of those scenarios is performed via k-means clustering using the scalar representations of the hydrometeorological forcing conditions proposed in (Rohmer et al., 2020) as inputs. A k-means algorithm considering 20 clusters is applied where the index (number of scenarios) of the closest point (set of scalar hydrometeorological conditions) to each cluster is retained to construct the learning set of scenarios. This leads to an initial set of $R_o = 20$ learning scenarios. To enrich the learning dataset, new scenarios are added by repeating the same procedure but applying the k-means algorithm to the scenarios that have not been previously chosen. Therefore, for each addition, we have that $R_i = R_{i-1} + \Delta R_i$, for $i = 1, \ldots, N_R$, where ΔR_i is the number of added scenarios and N_R is the number of enrichments. After defining the learning sets of

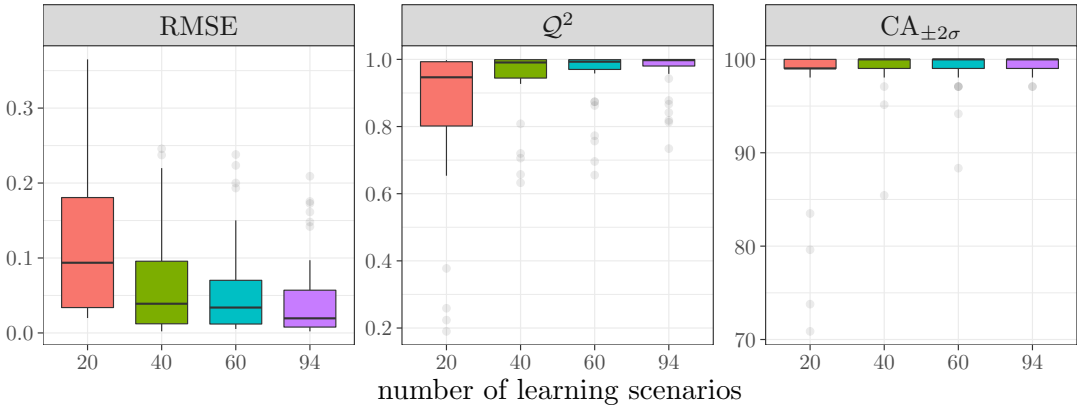


Figure 13: RMSE (left), $\mathcal{Q}^2$ (middle) and $\text{CA}_{\pm 2\sigma}$ (right) results for the experiment in Section 3.4.3. The boxplots are computed over the first 37 scenarios of the dataset in Section 2.1. The results are shown for different numbers of learning scenarios. The learning scenarios are taken only from the remaining 94 simulated scenarios of the coastal flooding dataset.

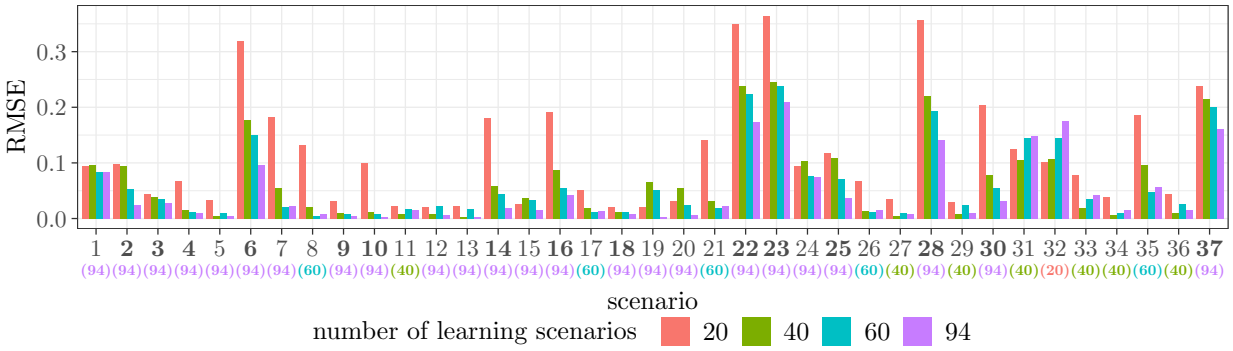


Figure 14: RMSE results of predictions for the first 37 scenarios of the dataset in Section 2.1 and considering $R = 20, 40, 60, 94$. The bars represent the magnitude of the RMSE criterion for each scenario. The fifteen scenarios where the RMSEs improved at each of the three increases of R are in bold. The subindices at the bottom of the x-axis designate which case provided the smaller RMSE.

flood scenarios, we then fit the corresponding GP models for the resulting 37 test cases. Both the training and prediction steps are performed considering the spatial design points proposed in Section 3.4.2.

Boxplots of the RMSE, $\mathcal{Q}^2$ and $\text{CA}_{\pm 2\sigma}$ values of the 37 predictions and considering $R = 20, 40, 60, 94$ learning scenarios are shown in Figure 13. Improvements are obtained on the three performance indicators as R increases, leading to smaller dispersion of the boxplots. For each test scenario, the $\text{CA}_{\pm 2\sigma}$ remains satisfactory regardless of R, and the RMSE and $\mathcal{Q}^2$ results are commonly outperformed when considering a larger learning dataset. More precisely, 24 of the 37 predictions are improved by considering all 94 learning scenarios. However, for some cases (see Figure 14, e.g., the RMSE results for scenarios 31 and 32), the prediction quality decreased.

We first analyze results for the 32nd scenario. By comparing the similarity between the hydrometeorological conditions using either 20 or 94 learning scenarios, the strongest correlations are provided with scenarios 64 and 68, respectively (Figure 15(a)). For the ease of the discussion, we focus on the comparison of the still water level (SWL), which is given by $\text{SWL}(t) = \text{MSL}(t) + \text{T}(t) + \text{S}(t)$.[9] Figure 15(b) shows that the SWL of scenario 68 is actually closer to that of scenario 32. Therefore, after adding scenario 68 to the learning dataset (event already added for $R = 60$), models may consider stronger correlations with respect to this scenario rather than the 64th scenario. However, while scenarios 32 and 64 correspond to moderate flood events, the 68th scenario corresponds to a weaker flood event (Figure 15(c)), explaining the misprediction when considering such scenarios in the learning dataset.

[9]By performing sensitivity analysis (SA) based on GPs, models were more sensitive to the MSL, T and S rather than the other hydrometeorological drivers. For the SA, the `sensitivity` package (Iooss et al., 2020) was adapted to account for functional data.

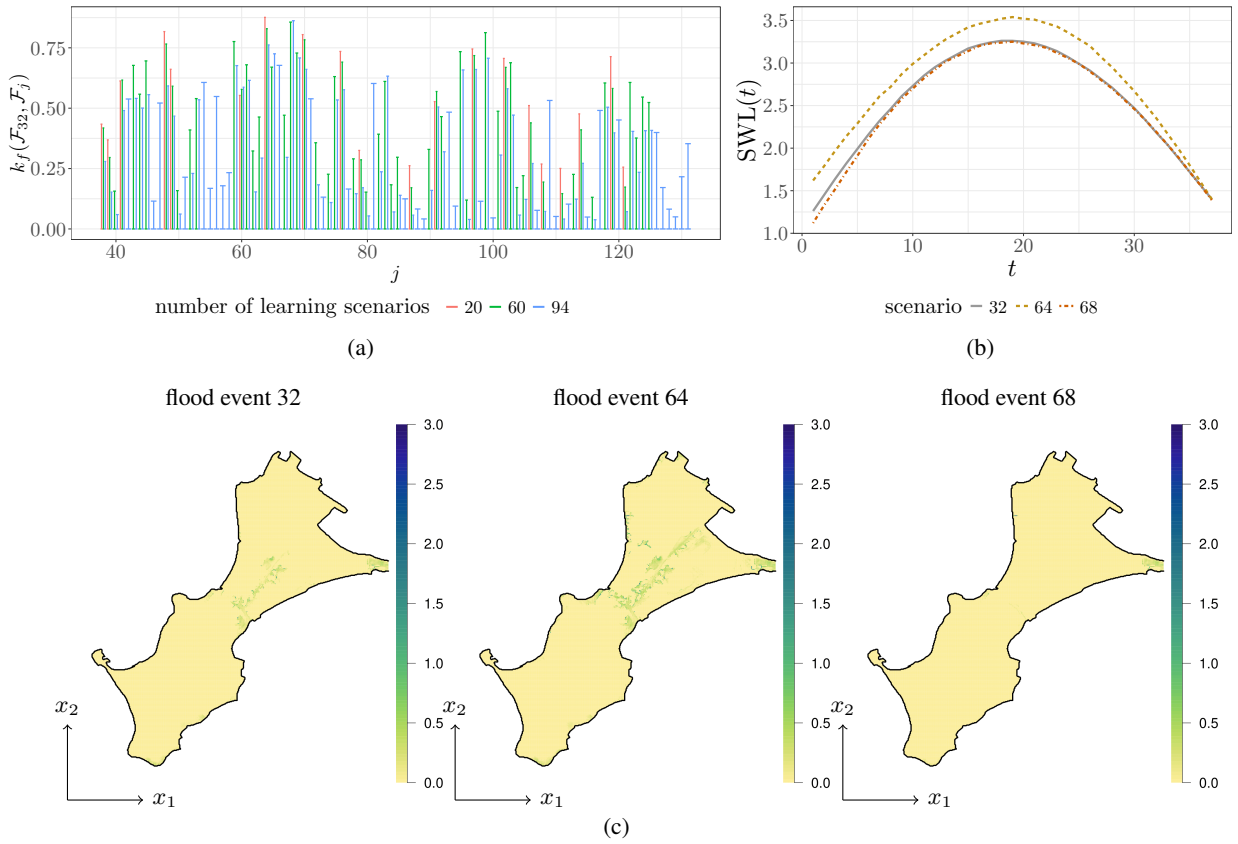


Figure 15: (a) Correlations between the hydrometeorological conditions of scenario 32 and those of the learning set. The results considering different numbers of learning scenarios are shown. (b) Time series of the still water level (SWL(t) = MSL(t) + T(t) + S(t)) for scenarios 32, 64 and 68. (c) True $H_{\max}$ maps of the scenarios under analysis.

Similar to scenario 32, similar behaviors were observed in other scenarios where the RMSE systematically decreased as R increased. To avoid this type of drawback, we can enrich the learning dataset with additional flood scenarios and/or consider adding complementary hydrometeorological conditions (or prior information on flood events) as inputs of the GP model that may help to discriminate scenarios. Figure 15(b) shows that small variations in the hydrometeorological conditions may lead to events with and without floods. Indeed, this remind us that floods occur only if the instantaneous water level resulting from the joint action of the MSL, tide, surge, waves and wind exceeds the level of the (natural or man-made) coastal defense crests. In that case, our framework must instead be adapted to learn the critical forcing conditions leading the instantaneous water level to exceed the defense crests.

3.4.4 Influence of the number of spatial design points

We now assess the impact of the number of spatial design points S in predictions. To do so, we repeat the experiment in Section 3.4.2 but consider 900 additional design points per flood event, i.e., $S = 1003$. This leads to a total of $N \sim 1.3 \times 10^5$ spatial design points. The same parameterization proposed in Section 3.4.2 is used here. Figure 16 shows the predictions for scenarios 43 and 100. As also observed in Figure 11, the model provides accurate predictions, leading to $\mathcal{Q}^2$ and CA$_{\pm 2\sigma}$ values above 85% and 99%, respectively. In practice, we may be interested in predicting flood areas rather than locally predicting values only at the subset of S spatial locations. This can be achieved by applying the conditional predictive GP formulas to the spatial domain of interest (see the discussion in Sections 2.2 and 2.3). Predictions of scenarios 1, 43 and 100 considering only spatial locations with nonzero EFPs ($\sim 34 \times 10^3$ points) are shown in Figure 17 for $S = 103, 1003$. For both cases, our framework tends to capture the intensity of those flood events. While spatial profiles are smoother by considering $S = 103$, a better predictability resolution is obtained for $S = 1003$. This is caused by the difference in the magnitude of the estimated length-scales $(\ell_{x,1}, \ell_{x,2})$ [m]. This should remind us that for larger values of $(\ell_{x,1}, \ell_{x,2})$, models lead to stronger spatial correlations and therefore to smoother predictions. Here, while the former GP model with $S = 103$ resulted in length-scale (median) estimations equal to

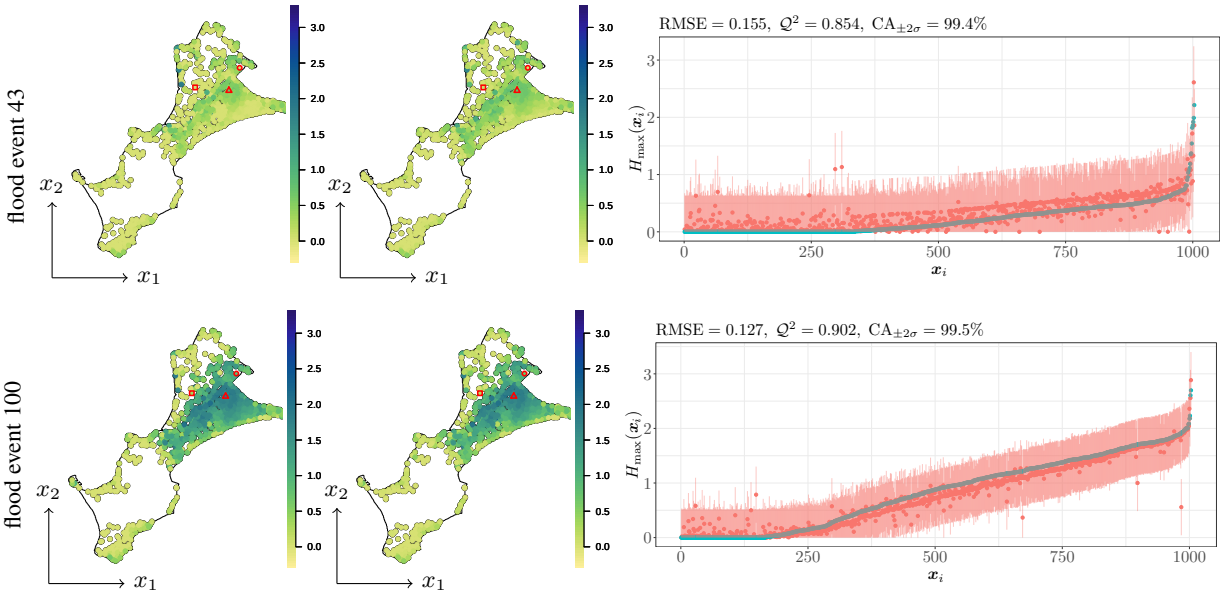


Figure 16: The $H_{\max}$ predictions for the experiments in Figure 11 considering 1003 spatial design points per flood scenario. The panel description is the same as that in Figure 11.

Table 3: Quality assessment of predictions in Figure 17 considering the flood categories recommended by the French Risk Prevention Plan. Flood events are classified as minor ($H_{\max} \leq 0.5$), moderate ($0.5 < H_{\max} \leq 1$), serious ($1 < H_{\max} \leq 1.5$) and severe ($H_{\max} > 1.5$). The proportions [%] per category are computed using spatial locations with nonzero EFPs. The results are shown considering the true observations ($H_{\max} := H$) and predictions using either $S_1 = 103$ ($\widehat{H}_{S_1}$) or $S_2 = 1003$ ($\widehat{H}_{S_2}$). The closest proportions to the ones provided by H (given in gray) are highlighted in bold.

Flood Category	Proportions [%] per Category								
	Scenario 1			Scenario 43			Scenario 100		
	H	$\widehat{H}_{S_1}$	$\widehat{H}_{S_2}$	H	$\widehat{H}_{S_1}$	$\widehat{H}_{S_2}$	H	$\widehat{H}_{S_1}$	$\widehat{H}_{S_2}$
minor	99.9	**99.7**	99.6	90.9	**86.6**	83.3	50.6	51.3	**50.8**
moderate	0.1	**0.3**	**0.3**	7.9	**12.0**	15.4	16.1	**17.3**	19.9
serious	0.0	**0.0**	0.1	0.9	0.8	**0.9**	20.0	**20.9**	18.0
severe	0.0	**0.0**	**0.0**	0.3	0.6	**0.4**	13.4	10.5	**11.2**

$\hat{\ell}_{x,1,103} = 42.7$ and $\hat{\ell}_{x,2,103} = 135.1$, the latter model with $S = 1003$ resulted in smaller length-scales: $\hat{\ell}_{x,1,1003} = 19.1$ and $\hat{\ell}_{x,2,1003} = 28.4$. For $S = 1003$, overestimations commonly occur in the areas surrounding buildings (small white zones). Indeed, the model overestimates $H_{\max}$ values at the center of buildings (locations that were never observed in the training step; see the discussion in Section 3.4.1) where those values are actually equal to zero. Therefore, because of the smoothness condition of GPs, predictions in the buildings' neighborhood are affected.

Table 3 assesses the quality of the predictions in Figure 17 according to the flood categories recommended by the French Risk Prevention Plan (Direction Générale de la Prévention des Risques, 2014). In this plan, depending on the impact of the flood (e.g., the water height of the neighboring district and/or in the road network), events are classified as follows: minor if $H_{\max} \leq 0.5$, moderate if $0.5 < H_{\max} \leq 1$, serious if $1 < H_{\max} \leq 1.5$ and severe if $H_{\max} > 1.5$. Then, for scenarios in Figure 17, we pointwise assign predictions $\widehat{H}_{\max}$ to their corresponding categories, and we compare the resulting proportions [%] per flood category with respect to those led by the true observations $H_{\max}$. From Table 3, we can observe that both models, considering $S = 103$ or $S = 1003$, globally lead to reliable flood proportions. Misclassification between consecutive categories mainly results from the overestimation or underestimation around the $H_{\max}$ threshold that defines the limit of each category. As an example, for scenario 43, a significant number of $H_{\max}$ values close to 0.5 were assigned in the moderate category when they actually belong to the minor category.

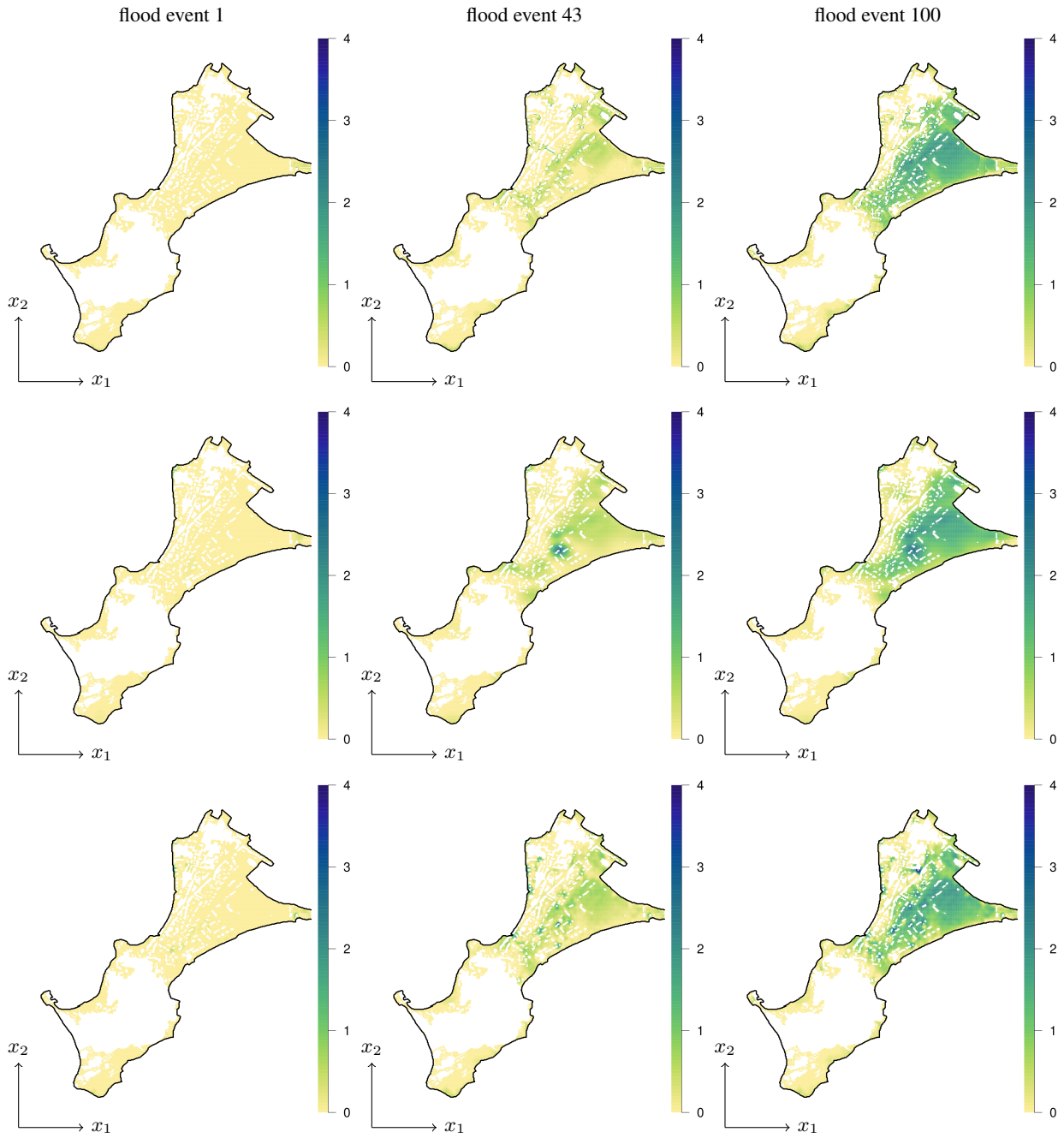


Figure 17: Predictions for the experiment in Section 3.4.2 considering only spatial locations with nonzero EFPs. The panels show the true $H_{\max}$ maps (top) and the predictions using either 103 (middle) or 1003 design points per flood event (bottom).

Finally, Figure 18 shows the boxplots of the performance indicator values for the LOO predictions and considering $S = 103, 503, 1003$. As discussed in Sections 3.4.2 and 3.4.3, to avoid misprediction due to the uniqueness of some scenarios, the analysis is focused on the first 37 flood events. The indicators (RMSE, $\mathcal{Q}^2$, and $\text{CA}_{\pm 2\sigma}$) are computed considering only spatial locations with nonzero EFPs (see Figure 10). They improve as S increases, leading to smaller dispersion of the boxplots. In terms of the computational costs, note that although the CPU times for both training and prediction steps increase as S increases, they remain tractable.[10] In practice, since the training step is executed only once and can be computed offline, we only need to be aware of the computational costs of the prediction step. Here,

[10]These experiments were executed on a single core of an HP cluster with an AMD quadprocessor and 256 GB RAM.

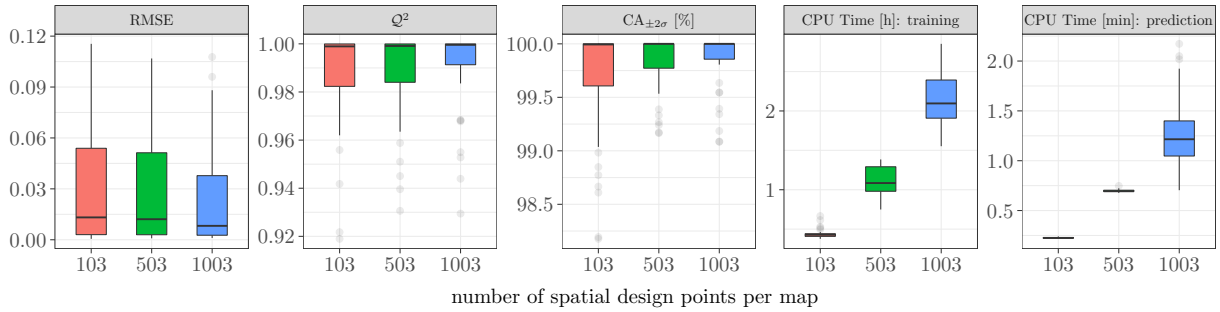


Figure 18: Performance indicators for the experiment in Section 3.4.4 considering the first 37 scenarios of the dataset in Section 2.1. The results are shown for $S = 103, 503, 1003$.

the prediction of a single map takes a couple of minutes, an advantage compared to the couple of days required by numerical simulators. Hence, this makes possible the use of our framework for FEWSs.

4 Conclusions

In this paper, we established a GP-based surrogate model that accounts for the effect of time-varying forcing conditions (e.g., tides, storms, surges, etc.) and provides spatially varying inland flood information (maximal inland water height, $H_{\max}$). We demonstrated that our framework can be easily applied without any parameterization of the hydrometeorological drivers in scalar representations, a step that is not always simple and that can lead to misleading predictions. To the best of our knowledge, our proposal is the first to consider both inputs and outputs of a coastal computer code as functions.

The proposed GP model was built on the proposition of a separable kernel that correlates both hydrometeorological forcing conditions (functional inputs) and spatial input locations. The resulting process can be seen as a multioutput GP where correlated spatial outputs are driven by multiple hydrometeorological functional inputs. Efficient implementations were explored by considering Kronecker-structured operations and/or sparse-variational approximations. This led to models that run fast for up to hundreds of outputs and thousands of spatial design points. Both Python and R codes were provided.

Our framework was widely tested on various examples considering different situations depending on data availability. According to the numerical experiments, our approach led to accurate predictions that outperformed standard GP implementations that cannot account for functional data. For forecasting unobserved spatial events, we numerically showed the convergence of our approach as the number of learning flood events increased. We demonstrated that accurate predictions can be obtained within tractable time lapses in the coastal flooding application. More precisely, predictions were obtained on the order of minutes compared to the couple of days required by dedicated simulators. This is key for FEWSs. We must note that the performance of the proposed framework depends on the availability and diversity of learning flood scenarios. The richer and diverse the learning dataset is, the better the predictability.

The framework presented here can be improved in different ways. A natural extension relies on projecting the spatial outputs onto (truncated) basis representations such as wavelets. In that case, functional inputs can be treated as discussed in this paper, and Gaussian priors will be placed over the space of the coefficients of representations rather than the output space. This would lead to significant improvements in terms of the computational costs for applications with large learning datasets, e.g., involving thousands of outputs and tens of thousands of design points. The nature of coastal flooding data, more precisely, the nonnegative, zero-inflated and discontinuous patterns (as illustrated by Figure 2), cannot be easily learned using purely GP assumptions. Hence, alternative (nonstationary) GP-based priors may be explored to obtain more realistic surrogate models.

Acknowledgement

This work was funded by the ANR (French National Research Agency) under the RISCOPE project (ANR-16-CE04-0011): https://perso.math.univ-toulouse.fr/riscope/. We thank J. Betancourt (IMT) for his helpful recommendations and for providing 94 of the 131 hydrometeorological scenarios. We also thank F. Gamboa (IMT, ANITI) and T. Klein (IMT, ENAC) for their fruitful discussions within the RISCOPE project. Part of this research was conducted when AFLL was a postdoctoral researcher at IMT and BRGM.

A Projection of Functional Inputs onto PCA Basis Functions

Consider the τ-length functional vector $\boldsymbol{f} = [f(t_1), \ldots, f(t_\tau)]^\top$. Let $\boldsymbol{F} = [\boldsymbol{f}_1, \ldots, \boldsymbol{f}_N]^\top$ be a matrix containing N replicates of $\boldsymbol{f}$. The PCA of $\boldsymbol{F}$ can be obtained using the variance-covariance matrix (Ramsay and Silverman, 2005). Denote the matrix $\boldsymbol{F}_c$ as the centered version of $\boldsymbol{F}$, where the mean of each column of $\boldsymbol{F}$ is subtracted from the column. Then, the variance-covariance matrix is given by

$$\boldsymbol{\Omega}_\tau = \frac{1}{N}\boldsymbol{F}_c^\top \boldsymbol{F}_c = \sum_{j=1}^{\tau} \lambda_j \boldsymbol{\nu}_j \boldsymbol{\nu}_j^\top, \tag{10}$$

where $\lambda_1 \geq \cdots \geq \lambda_\tau$ and $\boldsymbol{\nu}_1, \ldots, \boldsymbol{\nu}_\tau$ are the corresponding eigenvalues and eigenvectors of $\boldsymbol{\Omega}_\tau$, respectively. In practice, (10) is commonly truncated after reaching a predefined inertia $\mathcal{I}_*$, where the contribution of each eigencomponent is given by

$$\mathcal{I}_j = \frac{\lambda_j}{\sum_{\kappa=1}^{\tau} \lambda_\kappa}.$$

Then, the "optimal" number of PCA components is obtained by the smallest value of τ_* such that $\sum_{j=1}^{\tau_*} \mathcal{I}_j \geq \mathcal{I}_*$, where $\tau_* \leq \tau$. Finally, the eigenvectors of the truncated matrix $\boldsymbol{\Omega}_{\tau_*}$ can be used as basis functions in the linear approximation in (4), i.e., $[\phi_{i,1}, \ldots, \phi_{i,p_i}] = [\boldsymbol{\nu}_1, \ldots, \boldsymbol{\nu}_{\tau_*}]$ where $p_i = \tau_*$. Due to the orthonormality of $\boldsymbol{\nu}_1, \ldots, \boldsymbol{\nu}_{\tau_*}$, the vector of coefficients $\boldsymbol{\alpha}_\kappa$, for $\kappa = 1, \ldots, N$, associated with the replicate $\boldsymbol{f}_\kappa$ is given by $\boldsymbol{\alpha}_\kappa = \boldsymbol{\Phi}^\top \boldsymbol{f}_\kappa$ with $\boldsymbol{\Phi} = [\boldsymbol{\nu}_1, \ldots, \boldsymbol{\nu}_{\tau_*}]$.

B Kronecker-based Operations for Gaussian Processes with Separable Kernels

For large datasets, computing the covariance matrix $\mathbf{K} = (\mathbf{K}_f \otimes \mathbf{K}_x) \in \mathbb{R}^{RS \times RS}$ with R outputs and S spatial design points per output may easily run out the memory. This limitation exists when constructing the (lower triangular) Cholesky factorization $\mathbf{L} = (\mathbf{L}_f \otimes \mathbf{L}_x) \in \mathbb{R}^{RS \times RS}$ and its inverse. To mitigate this drawback, instead of computing $\mathbf{K}$ or $\mathbf{L}$, we propose solving triangular-structured linear systems involving Kronecker products.

B.1 Properties of the Kronecker product

We first recall the properties of the Kronecker product that are needed in this appendix (see Laub, 2004; Alvarez et al., 2012, for further details). Consider the matrices $\mathbf{A}, \mathbf{A}' \in \mathbb{R}^{M \times N}, \mathbf{B}, \mathbf{B}' \in \mathbb{R}^{P \times Q}$ and the vectors $\boldsymbol{u} \in \mathbb{R}^{NQ}, \boldsymbol{v} \in \mathbb{R}^{MP}$. Then, some useful properties of the Kronecker product are:

$$\begin{aligned} (\mathbf{A} \otimes \mathbf{B})^\top &= \mathbf{A}^\top \otimes \mathbf{B}^\top, \\ (\mathbf{A} \otimes \mathbf{B})^{-1} &= \mathbf{A}^{-1} \otimes \mathbf{B}^{-1}, \\ (\mathbf{A} \otimes \mathbf{B})(\mathbf{A}' \otimes \mathbf{B}') &= (\mathbf{A}\mathbf{A}' \otimes \mathbf{B}\mathbf{B}'). \end{aligned} \tag{11}$$

We are also interested in computing (or solving) linear systems of the form:

$$\boldsymbol{v} = (\mathbf{A} \otimes \mathbf{B})\boldsymbol{u}. \tag{12}$$

For computing (12), note that it is possible to rearrange the vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ in matrices in order to avoid constructing $(\mathbf{A} \otimes \mathbf{B})$ (a step that required computing and storing an $MP \times NQ$ matrix). Consider the matrices $\mathbf{U} \in \mathbb{R}^{Q \times N}$ and $\mathbf{V} \in \mathbb{R}^{P \times M}$, where the elements of $\boldsymbol{u}$ and $\boldsymbol{v}$ are indexed by columns. Then, (12) is given by

$$\mathbf{V} = \mathbf{B}\mathbf{U}\mathbf{A}^\top. \tag{13}$$

After computing $\mathbf{V}$, which involves computing and storing smaller matrices, we can obtain $\boldsymbol{v}$ by vectorizing $\mathbf{V}$.

B.2 Computation of the likelihood

We must note that the complexity of the Gaussian likelihood relies on the computation of the quadratic term:

$$z = \boldsymbol{y}^\top \mathbf{K}^{-1} \boldsymbol{y}, \tag{14}$$

with covariance matrix $\mathbf{K} = \mathbf{K}_x \otimes \mathbf{K}_f$ and an observation vector $\boldsymbol{y} = [y_1, \ldots, y_N]^\top$ with $N = SR$. Note that to achieve numerical simplifications in further steps, we change the order of the Kronecker product proposed in (6) since commonly $R \ll S$. This only implies properly rearranging the indexation of the observations in $\boldsymbol{y}$, i.e..,

$$\boldsymbol{y} = [Y(\boldsymbol{\mathcal{F}}_1, \boldsymbol{x}_1), \ldots, Y(\boldsymbol{\mathcal{F}}_1, \boldsymbol{x}_S), \ldots, Y(\boldsymbol{\mathcal{F}}_R, \boldsymbol{x}_1), \ldots, Y(\boldsymbol{\mathcal{F}}_R, \boldsymbol{x}_S)]^\top.$$

Then, using the Cholesky factorization of $\mathbf{K}$, we can show that (14) is given by

$$z = \boldsymbol{y}^\top (\mathbf{L}\mathbf{L}^\top)^{-1}\boldsymbol{y} = (\mathbf{L}^{-1}\boldsymbol{y})^\top(\mathbf{L}^{-1}\boldsymbol{y}) = \boldsymbol{a}^\top\boldsymbol{a}.$$

For computing $\boldsymbol{a} = \mathbf{L}^{-1}\boldsymbol{y} = (\mathbf{L}_x^{-1} \otimes \mathbf{L}_f^{-1})\boldsymbol{y} \in \mathbb{R}^N$, by using (13), we then have:

$$\mathbf{A} = \mathbf{L}_f^{-1}\mathbf{Y}(\mathbf{L}_x^{-1})^\top = ({\mathbf{L}_x}^{-1}[{\mathbf{L}_f}^{-1}\mathbf{Y}]^\top)^\top = ({\mathbf{L}_x}^{-1}\mathbf{B}^\top)^\top, \tag{15}$$

where $\mathbf{B} = {\mathbf{L}_f}^{-1}\mathbf{Y}$. Hence, to construct the matrix $\mathbf{B} \in \mathbb{R}^{R\times N}$, we need to solve the linear system $\mathbf{L}_f\mathbf{B} = \mathbf{Y}$, which can be efficiently computed since $\mathbf{L}_f$ is a lower triangular matrix. Similarly, matrix $\mathbf{A} \in \mathbb{R}^{R\times S}$ is obtained by solving the triangular-structured system $\mathbf{L}_x\mathbf{A}^\top = \mathbf{B}^\top$. Finally, vector $\boldsymbol{a}$ results from vectorizing the matrix $\mathbf{A}$.

Note that for computing (14) using (15), intermediate steps require constructing and storing $R \times N$ and $R \times S$ matrices rather than the inverse of the $N \times N$ covariance matrix. In our application, this led to significant computational improvements since we considered large numbers of spatial design points S.

B.3 Computation of the conditional mean and covariance functions

From (2), we can note that the conditional mean function μ and the conditional covariance function c also depend on the computation of $\mathbf{K}$. As in Appendix B.2, here, we provide efficient computations of those quantities by exploiting the properties of the Kronecker product (see Appendix B.1).

We first focus on the computation of μ. For the ease of the notation, we denote $\mu := \mu(\boldsymbol{x}_*, \boldsymbol{\mathcal{F}}_*)$ and

$$\boldsymbol{k} := \boldsymbol{k}(\boldsymbol{x}_*, \boldsymbol{\mathcal{F}}_*) = [k((\boldsymbol{x}_*, \boldsymbol{\mathcal{F}}_*), (\boldsymbol{x}_1, \boldsymbol{\mathcal{F}}_1)), \ldots, k((\boldsymbol{x}_*, \boldsymbol{\mathcal{F}}_*), (\boldsymbol{x}_N, \boldsymbol{\mathcal{F}}_N))]^\top.$$

Then, following a similar procedure as the one used in Appendix B.2 and using (11), we can show that μ is given by

$$\mu = \boldsymbol{k}^\top\mathbf{K}^{-1}\boldsymbol{y} = \boldsymbol{k}^\top({\mathbf{L}_x}^\top \otimes {\mathbf{L}_f}^\top)^{-1}\boldsymbol{a},$$

with $\boldsymbol{a} = (\mathbf{L}_x^{-1} \otimes \mathbf{L}_f^{-1})\boldsymbol{y}$. Now, since $\boldsymbol{k} = \boldsymbol{k}_x \otimes \boldsymbol{k}_f$ with $\boldsymbol{k}_x = [k_x(\boldsymbol{x}_*, \boldsymbol{x}_1), \ldots, k_x(\boldsymbol{x}_*, \boldsymbol{x}_S)]^\top$ and $\boldsymbol{k}_f = [k_f(\boldsymbol{\mathcal{F}}_*, \boldsymbol{\mathcal{F}}_1), \ldots, k_f(\boldsymbol{\mathcal{F}}_*, \boldsymbol{\mathcal{F}}_R)]^\top$, (11) yields:

$$\mu = ([{\mathbf{L}_x}^{-1}\boldsymbol{k}_x]^\top \otimes [{\mathbf{L}_f}^{-1}\boldsymbol{k}_f]^\top)\boldsymbol{a}. \tag{16}$$

As discussed in Appendix B.2, the computations $\boldsymbol{b}_x = {\mathbf{L}_x}^{-1}\boldsymbol{k}_x$ and $\boldsymbol{b}_f = {\mathbf{L}_f}^{-1}\boldsymbol{k}_f$ are efficient since $\mathbf{L}_x$ and $\mathbf{L}_f$ are lower triangular matrices. Then, (16) is obtained by applying (13).

Now, denote $c := c((\boldsymbol{x}_*, \boldsymbol{\mathcal{F}}_*), (\boldsymbol{x}'_*, \boldsymbol{\mathcal{F}}'_*))$, $k := k((\boldsymbol{x}_*, \boldsymbol{\mathcal{F}}_*), (\boldsymbol{x}'_*, \boldsymbol{\mathcal{F}}'_*))$ and $\boldsymbol{k}' := \boldsymbol{k}(\boldsymbol{x}'_*, \boldsymbol{\mathcal{F}}'_*) = [k((\boldsymbol{x}_1, \boldsymbol{\mathcal{F}}_1), (\boldsymbol{x}'_*, \boldsymbol{\mathcal{F}}'_*)), \ldots, k((\boldsymbol{x}_N, \boldsymbol{\mathcal{F}}_N), (\boldsymbol{x}'_*, \boldsymbol{\mathcal{F}}'_*))]^\top$. Following a similar procedure as the one used for μ, we have that c is given by

$$c = k - \boldsymbol{k}^\top\mathbf{K}^{-1}\boldsymbol{k}' = k - (\boldsymbol{b}_x \otimes \boldsymbol{b}_f)^\top(\boldsymbol{b}'_x \otimes \boldsymbol{b}'_f) = k - (\boldsymbol{b}_x^\top\boldsymbol{b}'_x)(\boldsymbol{b}_f\boldsymbol{b}'_f), \tag{17}$$

with $\boldsymbol{b}_x = {\mathbf{L}_x}^{-1}\boldsymbol{k}_x$, $\boldsymbol{b}_f = {\mathbf{L}_f}^{-1}\boldsymbol{k}_f$, $\boldsymbol{b}'_x = {\mathbf{L}_x}^{-1}\boldsymbol{k}'_x$, and $\boldsymbol{b}_f = {\mathbf{L}_f}^{-1}\boldsymbol{k}'_f$.

References

Alvarez, M. A., Rosasco, L., and Lawrence, N. D. (2012). Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, 4(3).

André, C., Monfort, D., Bouzit, M., and Vinchon, C. (2013). Contribution of insurance data to cost assessment of coastal flood damage to residential buildings: Insights gained from Johanna (2008) and Xynthia (2010) storm events. *Natural Hazards and Earth System Sciences*, 13(8):2003–2012.

Azzimonti, D., Ginsbourger, D., Rohmer, J., and Idier, D. (2019). Profile extrema for visualizing and quantifying uncertainties on excursion regions. Application to coastal flooding. *Technometrics*, 0(ja):1–26.

Bathrellos, G. D., Skilodimou, H. D., Chousianitis, K., Youssef, A. M., and Pradhan, B. (2017). Suitability estimation for urban development using multi-hazard assessment map. *Science of The Total Environment*, 575:119–134.

Bertin, X., Bruneau, N., Breilh, J. F., Fortunato, A., and Karpytchev, M. (2012). Importance of wave age and resonance in storm surges: The case Xynthia, Bay of Biscay. *Ocean Modelling*, 42:16–30.

Betancourt, J., Bachoc, F., Klein, T., Idier, D., Pedreros, R., and Rohmer, J. (2020). Gaussian process metamodeling of functional-input code for coastal flood hazard assessment. *Reliability Engineering & System Safety*, 198:106870.

Bilionis, I., Zabaras, N., Konomi, B. A., and Lin, G. (2013). Multi-output separable Gaussian process: Towards an efficient, fully Bayesian paradigm for uncertainty quantification. *Journal of Computational Physics*, 241:212–239.

Blake, E. S., Rappaport, E. N., Jarrell, J. D., and Landsea, C. W. (2006). The deadliest, costliest, and most intense United States tropical cyclones from 1851 to 2005 (and other frequently requested hurricane facts). Technical Report NOAA Technical Memorandum NWS TPC-4, Syst. Dev. Off., National Weather Service, Silver Spring.

Bolle, A., , Neves, L., Smets, S., Mollaert, J., and Buitrago, S. (2018). An impact-oriented Early Warning and Bayesian-based decision support system for flood risks in Zeebrugge harbour. *Coastal Engineering*, 134:191–202.

Bouhlel, M. A., Hwang, J. T., Bartoli, N., Lafage, R., Morlier, J., and Martins, J. (2019). A Python surrogate modeling framework with derivatives. *Advances in Engineering Software*, page 102662.

Camps-Valls, G., Verrelst, J., Munoz-Mari, J., Laparra, V., Mateo-Jimenez, F., and Gomez-Dans, J. (2016). A survey on Gaussian processes for earth-observation data analysis: A comprehensive investigation. *IEEE Geoscience and Remote Sensing Magazine*, 4(2):58–78.

Cao, Q. D., Miles, S. B., and Choe, Y. (2021). Infrastructure recovery curve estimation using Gaussian process regression on expert elicited data. *Reliability Engineering & System Safety*, page 108054.

Chang, K. and Guillas, S. (2019). Computer model calibration with large non-stationary spatial outputs: application to the calibration of a climate model. *Journal of the Royal Statistical Society Series C*, 68(1):51–78.

Da Veiga, S. and Marrel, A. (2020). Gaussian process regression with linear inequality constraints. *Reliability Engineering & System Safety*, 195:106732.

Deville, Y., Ginsbourger, D., and Roustant, O. (2019). *kergp: Gaussian Process Laboratory*. R package version 0.5.0.

Direction Générale de la Prévention des Risques (2014). *Guide méthodologique : Plan de prévention des risques littoraux* (in french). Technical report, Service des Risques Naturels et Hydraulique.

Doong, D.-J., Chuang, L. Z.-H., Wu, L.-C., Fan, Y.-M., Kao, C. C., and Wang, J.-H. (2012). Development of an operational coastal flooding early warning system. *Natural Hazards and Earth System Sciences*, 12(2):379–390.

Fricker, T. E., Oakley, J. E., and Urban, N. M. (2013). Multivariate Gaussian process emulators with nonseparable covariance structures. *Technometrics*, 55(1):47–56.

Fukutani, Y., Moriguchi, S., Terada, K., and Otake, Y. (2021). Time-dependent probabilistic tsunami inundation assessment using mode decomposition to assess uncertainty for an earthquake scenario. *Journal of Geophysical Research: Oceans*, 126(7):e2021JC017250.

Genton, M. G. (2001). Classes of kernels for machine learning: A statistics perspective. *Journal of Machine Learning Research*, 2:299–312.

Górecki, T. and Piasecki, P. (2019). A comprehensive comparison of distance measures for time series classification. In Steland, A., Rafajłowicz, E., and Okhrin, O., editors, *Stochastic Models, Statistics and Their Applications*, pages 409–428, Cham. Springer International Publishing.

Hartigan, J. A. and Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108.

Hensman, J., Fusi, N., and Lawrence, N. D. (2013). Gaussian processes for big data. In *Conference on Uncertainty in Artificial Intelligence*, volume 29, pages 282–290.

Hoggart, S., Hanley, M., Parker, D., Simmonds, D., Bilton, D., Filipova-Marinova, M., Franklin, E., Kotsev, I., Penning-Rowsell, E., Rundle, S., Trifonova, E., Vergiev, S., White, A., and Thompson, R. (2014). The consequences of doing nothing: The effects of seawater flooding on coastal zones. *Coastal Engineering*, 87:169–182.

Idier, D., Aurouet, A., Bachoc, F., Baills, A., Betancourt, J., Durand, J., Mouche, R., Rohmer, J., Gamboa, F., Klein, T., Lambert, J., Cozannet, G. L., Roy, S. L., Louisor, J., Pedreros, R., and Véron, A.-L. (2020a). Toward a user-based, robust and fast running method for coastal flooding forecast, early warning, and risk prevention. *Journal of Coastal Research*, 95(sp1):1111–1116.

Idier, D., Rohmer, J., Bulteau, T., and Delvallée, E. (2013). Development of an inverse method for coastal risk management. *Natural Hazards and Earth System Sciences*, 13(4):999–1013.

Idier, D., Rohmer, J., Pedreros, R., Le Roy, S., Lambert, J., Louisor, J., Le Cozannet, G., and Le Cornec, E. (2020b). Coastal flood: A composite method for past events characterisation providing insights in past, present and future hazard-joining historical, statistical and modelling approaches. *Natural Hazards*, 101:465–501.

Iooss, B., Janon, A., and Pujol, G. (2020). *sensitivity: Global Sensitivity Analysis of Model Outputs*. R package version 1.18.0.

Iooss, B. and Ribatet, M. (2009). Global sensitivity analysis of computer models with functional inputs. *Reliability Engineering & System Safety*, 94(7):1194–1204. Special Issue on Sensitivity Analysis.

Jia, G. and Taflanidis, A. A. (2013). Kriging metamodeling for approximation of high-dimensional wave and surge responses in real-time storm/hurricane risk assessment. *Computer Methods in Applied Mechanics and Engineering*, 261–262:24–38.

Jin, R., Chen, W., and Sudjianto, A. (2005). An efficient algorithm for constructing optimal design of computer experiments. *Journal of Statistical Planning and Inference*, 134:268–287.

Johnson, S. G. (2020). The NLopt nonlinear-optimization package. http://ab-initio.mit.edu/nlopt.

Kim, S., Melby, J., Nadal-Caraballo, N., and Ratcliff, J. (2015). A time-dependent surrogate model for storm surge prediction based on an artificial neural network using high-fidelity synthetic hurricane modeling. *Natural Hazards*, 76(1):565–585.

Laub, A. J. (2004). *Matrix Analysis For Scientists And Engineers*. Society for Industrial and Applied Mathematics.

Lázaro-Gredilla, M. and Figueiras-Vidal, A. (2009). Inter-domain Gaussian processes for sparse inference using inducing features. In *Advances in Neural Information Processing Systems*, volume 22, pages 1087–1095.

Le Roy, S., Pedreros, R., André, C., Paris, F., Lecacheux, S., Marche, F., and Vinchon, C. (2015). Coastal flooding of urban areas by overtopping: Dynamic modelling application to the Johanna storm (2008) in Gâvres (France). *Natural Hazards and Earth System Sciences*, 15:2497–2510.

Li, M., Wang, R.-Q., and Jia, G. (2020). Efficient dimension reduction and surrogate-based sensitivity analysis for expensive models with high-dimensional outputs. *Reliability Engineering & System Safety*, 195:106725.

Liu, X. and Guillas, S. (2017). Dimension reduction for Gaussian process emulation: An application to the influence of bathymetry on tsunami heights. *SIAM/ASA Journal on Uncertainty Quantification*, 5(1):787–812.

Lumbroso, D. M. and Vinet, F. (2011). A comparison of the causes, effects and aftermaths of the coastal flooding of England in 1953 and France in 2010. *Natural Hazards and Earth System Sciences*, 11(8):2321–2333.

Marrel, A., Iooss, B., Jullien, M., Laurent, B., and Volkova, E. (2011). Global sensitivity analysis for models with spatially dependent outputs. *Environmetrics*, 22(3):383–397.

Marrel, A., Iooss, B., Laurent, B., and Roustant, O. (2009). Calculations of Sobol indices for the Gaussian process metamodel. *Reliability Engineering & System Safety*, 94(3):742–751.

Matthews, A. G. d. G., Van der Wilk, M., Nickson, T., Fujii, K., Boukouvalas, A., León-Villagrá, P., Ghahramani, Z., and Hensman, J. (2017). GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research*, 18(40):1–6.

Moreno-Muñoz, P., Artés, A., and Alvarez, M. A. (2018). Heterogeneous multi-output Gaussian process prediction. In *Advances in Neural Information Processing Systems*, volume 31, pages 6711–6720.

Mori, U., Mendiburu, A., and Lozano, J. A. (2016). Distance measures for time series in R: The TSdist package. *R J.*, 8:451.

Paciorek, C. J. and Schervish, M. J. (2004). Nonstationary covariance functions for Gaussian process regression. In *Advances in Neural Information Processing Systems*, volume 16, pages 273–280.

Perrin, T. V. E., Roustant, O., Rohmer, J., Alata, O., Naulin, J. P., Idier, D., Pedreros, R., Moncoulon, D., and Tinard, P. (2021). Functional principal component analysis for global sensitivity analysis of model with spatial output. *Reliability Engineering & System Safety*, 211:107522.

Ramsay, J. and Silverman, B. W. (2005). *Functional Data Analysis*. Springer Series in Statistics. Springer.

Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, Cambridge, MA.

Rohmer, J. and Idier, D. (2012). A meta-modelling strategy to identify the critical offshore conditions for coastal flooding. *Natural Hazards and Earth System Sciences*, 12(9):2943–2955.

Rohmer, J., Idier, D., Paris, F., Pedreros, R., and Louisor, J. (2018). Casting light on forcing and breaching scenarios that lead to marine inundation: Combining numerical simulations with a random-forest classification approach. *Environmental Modelling & Software*, 104:64–80.

Rohmer, J., Idier, D., and Pedreros, R. (2020). A nuanced quantile random forest approach for fast prediction of a stochastic marine flooding simulator applied to a macrotidal coastal site. *Stochastic Environmental Research and Risk Assessment*.

Rohmer, J., Lecacheux, S., Pedreros, R., Quetelard, H., Bonnardot, F., and Idier, D. (2016). Dynamic parameter sensitivity in numerical modelling of cyclone-induced waves: A multi-look approach using advanced meta-modelling techniques. *Natural Hazards*, 84(3):1765–1792.

Roustant, O., Padonou, E., Deville, Y., Clément, A., Perrin, G., Giorla, J., and Wynn, H. (2020). Group kernels for Gaussian process metamodels with categorical inputs. *SIAM/ASA Journal on Uncertainty Quantification*, 8(2):775–806.

Rueda, A., Cagigal, L., Pearson, S., Antolínez, J., Storlazzi, C., Van Dongeren, A., Camus, P., and Mendez, F. J. (2019). HyCReWW: A hybrid coral reef wave and water level metamodel. *Computers & Geosciences*, 127:85–90.

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989). Design and analysis of computer experiments. *Statistical Science*, 4(4):409–423.

Shi, J. and Choi, T. (2011). *Gaussian Process Regression Analysis for Functional Data*. Chapman and Hall/CRC, New York.

Stansby, P., Chini, N., Apsley, D., Borthwick, A., Bricheno, L., Horrillo-Caraballo, J., McCabe, M., Reeve, D., Rogers, B., Saulter, A., Scott, A., Wilson, C., Wolf, J., and Yan, K. (2013). An integrated model system for coastal flood prediction with a case history for Walcott, UK, on 9 November 2007. *Journal of Flood Risk Management*, 6(3):229–252.

Titsias, M. (2009). Variational learning of inducing variables in sparse Gaussian processes. In *Proceedings of Machine Learning Research*, volume 5, pages 567–574.

Tromble, E., Kolar, R., Dresback, K., Hong, Y., Vieux, B., Luettich, R., Gourley, J., Kelleher, K., and Van Cooten, S. (2019). Aspects of coupled hydrologic-hydrodynamic modeling for coastal flood inundation. In *Estuarine and Coastal Modeling*, pages 724–743.

Van der Wilk, M., Dutordoir, V., John, S., Artemev, A., Adam, V., and Hensman, J. (2020). A framework for interdomain and multioutput Gaussian processes. *arXiv preprint*.

Zhang, Y. and Edgar, T. (2008). PCA combined model-based design of experiments (DOE) criteria for differential and algebraic system parameter estimation. *Industrial & Engineering Chemistry Research*, 47:7772–7783.

Zijlema, M., Stelling, G., and Smit, P. (2011). SWASH: An operational public domain code for simulating wave fields and rapidly varied flows in coastal waters. *Coastal Engineering*, 58:992–1012.