

Tensor-variate Gaussian process regression for efficient emulation of complex systems: comparing regressor and covariance structures in outer product and parallel partial emulators.*

D. Semochkina[†], S. E. Jackson[‡], and D. C. Woods[†]

Abstract.

Multi-output Gaussian process regression has become an important tool in uncertainty quantification, for building emulators of computationally expensive simulators, and other areas such as multi-task machine learning. We present a holistic development of tensor-variate Gaussian process (TvGP) regression, appropriate for arbitrary dimensional outputs where a Kronecker product structure is appropriate for the covariance. We show how two common approaches to problems with two-dimensional output, outer product emulators (OPE) and parallel partial emulators (PPE), are special cases of TvGP regression and hence can be extended to higher output dimensions. Focusing on the important special case of matrix output, we investigate the relative performance of these two approaches. The key distinction is the additional dependence structure assumed by the OPE, and we demonstrate when this is advantageous through two case studies, including application to a spatial-temporal influenza simulator.

Key words. Emulation, Gaussian process regression, Separable covariance

MSC codes. 60G15, 62M30, 62P12, 62J99

1. Introduction. Scientific processes are commonly modeled using mathematical models implemented in computer codes, or simulators, that encapsulate the key features of the system and facilitate prediction and decision-making [9]. When these simulators are computationally expensive, it is common to approximate them using statistical emulators, most commonly a Gaussian process, constructed from computer experiments [40]. When the simulator output is dynamic, evolving over dimensions such as space or time, development of efficient and effective emulators is more challenging.

As motivation, we study a simulator of the spatio-temporal dynamics of an influenza epidemic, developed by the UK Health Security Agency (UKHSA). The code implements an SEIR model [22] that describes the time progression of a closed population through the states *Susceptible*, *Exposed*, *Infected* and *Recovered*. The model also has a discrete spatial dimension, made up of a set of “patches” representing different urban communities. Movement between these patches is controlled via a commuter matrix which simulates flows of people for work, education and leisure. While there is a well-understood, although intractable, set of differential equations governing this system, for the purposes of this paper we treat the simulator as a block box modeling the number of infected individuals and each simulator run having as output a matrix of spatio-temporal responses (numbers of infections for each time and patch).

Previous work [23, 12, 7, 35] on emulating simulators with dynamic or, relatedly, multivariate output has focused on vector-valued responses, where the output from each run is unstructured or the structure is preserved in, or collapsed onto, a single dimension. Most commonly, the covariance kernel for the Gaussian process emulator is formed as the product of two separate kernels describing, respectively, relationships between simulator inputs and simulator output indices. Two approaches which have seen particular application are (i) the Outer Product Emulator [39] (OPE), which extends the separability to include the mean function with a closely related model applied to spatio-temporal simulators [4]; and (ii) the Parallel Partial Emulator [18] (PPE), which assumes separate emulators across the dynamic output indices with common regression functions,

*Submitted to the editors 2024-12-12.

Funding: This work was supported by the Chemical and Biological Technologies Department (contract HDTRA1-17-C-0028) and the UK Engineering and Physical Sciences Research Council (grants EP/R009902/1 and EP/S019472/1).

[†]University of Southampton, UK (d.semochkina@soton.ac.uk, d.woods@soton.ac.uk).

[‡]Durham University, Durham, UK (samuel.e.jackson@durham.ac.uk)

but distinct regression parameters, and common kernel functions with shared parameters.

Motivated by work in the machine learning literature [45, 11, 46, 10], in this paper, we cast OPE and PPE as special cases of a tensor-variate Gaussian process (TvGP) which allows modeling of outputs of arbitrary dimension represented as tensors using separable covariance kernels (Section 2). We then perform an empirical comparison of the effectiveness of OPE and PPE via two case studies: the SEIR patch model described above (Section 3) and a synthetic environmental simulator [5] of a pollutant spillage from a chemical accident (Section 4). A primary aim of these studies is to assess the contribution of the different assumptions made for the two emulators, and to attempt to gain insight about situations when imposing beliefs about correlation structure is important. As such, a discussion of the results and the differences between the two emulators are also provided (Section 5 and Appendix B). Code for implementing the methods in this paper and reproducing the results is available at <https://github.com/samjacksonstats/TVGP>.

2. Multivariate Emulation.

2.1. Separable emulators. Consider a generic multivariate simulator, represented by the function \mathbf{f} , which takes a vector $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^p$ of input parameters, and outputs a vector $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^r$. We choose to represent our beliefs about the behavior of the simulator output for any input \mathbf{x} in the following form [16]:

$$(2.1) \quad \mathbf{f}(\mathbf{x}) = \boldsymbol{\mu}(\mathbf{x}) + \mathbf{e}(\mathbf{x}),$$

where $\boldsymbol{\mu} : \mathbb{R}^p \rightarrow \mathbb{R}^r$ is a vector-valued mean function, typically expressed as a linear predictor, and $\mathbf{e}(\mathbf{x})$ is an r -dimensional stochastic residual process aiming to capture residual variation across \mathcal{X} .

A prior belief specification across the collection $\{\boldsymbol{\mu}(\mathbf{x}), \mathbf{e}(\mathbf{x}) : \mathbf{x} \in \mathcal{X}\}$ leads to a prior specification about simulator output $\mathbf{f}(\mathbf{x})$ at any $\mathbf{x} \in \mathcal{X}$. Such specification is not trivial; however, it is common for the specification to assume $\mathbb{E}[\mathbf{e}(\mathbf{x})] = \mathbf{0}$, $\text{Cov}[\boldsymbol{\mu}(\mathbf{x}), \mathbf{e}(\mathbf{x})] = \mathbf{0}$, and a separable covariance structure between the value of $\mathbf{e}(\mathbf{x})$ and $\mathbf{e}(\mathbf{x}')$, at two inputs \mathbf{x} and \mathbf{x}' , of the following form [12]:

$$(2.2) \quad \text{Cov}[\mathbf{e}(\mathbf{x}), \mathbf{e}(\mathbf{x}')] = \kappa(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}_0) \Sigma.$$

This assumption has been shown to be very effective for constructing emulators for multivariate simulators with large r . Covariance between the output components at any given input is given by the product of the positive semi-definite coregionalization matrix $\Sigma \in \mathbb{R}^{r \times r}$ and the stationary kernel function $\kappa : \mathbb{R}^p \times \mathbb{R}^p \rightarrow \mathbb{R}$, parameterised by $\boldsymbol{\theta}_0$ (although we will drop the $\boldsymbol{\theta}_0$ in this expression from here-on in, assuming that such a parameterization is implicit). A common choice is the Gaussian correlation function [30, 3]:

$$(2.3) \quad \kappa(\mathbf{x}, \mathbf{x}') = \exp \left(- \sum_{h=1}^p \left\{ \frac{x_h - x'_h}{\theta_{0,h}} \right\}^2 \right),$$

which requires specification of correlation length parameters $\theta_{0,h}$, $h = 1, \dots, p$.

One particular specification of a prior stochastic process for $\mathbf{f}(\mathbf{x})$ with these properties is a multivariate Gaussian process (MvGP) [12, 35],

$$(2.4) \quad \mathbf{f}^T(\cdot) \sim \mathcal{MG}\mathcal{P}(\boldsymbol{\mu}^T(\cdot), \kappa(\cdot, \cdot), \Sigma).$$

This process is a multivariate generalisation for r -dimensional outputs of the usual one-dimensional Gaussian process [37], and captures many of the representations developed throughout the literature.

2.2. Tensor-variate Gaussian processes. There are several commonly observed structures in $\mathbf{f}(\mathbf{x})$, for example with entries corresponding to a spatial location in two dimensions or defined via space-time coordinates in higher dimensions. In Sections 3 and 4 we emulate simulators with two-dimensional space-time coordinates, that is, input \mathbf{x} produces outputs

$$\mathcal{F}(\mathbf{x}) = \begin{pmatrix} f_{11}(\mathbf{x}) & \dots & f_{1r_2}(\mathbf{x}) \\ \vdots & \ddots & \vdots \\ f_{r_1 1}(\mathbf{x}) & \dots & f_{r_1 r_2}(\mathbf{x}) \end{pmatrix}.$$

Here $f_{i_1 i_2}(\mathbf{x}) = f(\mathbf{x}; s_{i_1}, t_{i_2})$ represents model output for spatial grid point $s_{i_1} \in \mathcal{S}, i_1 = 1, \dots, r_1$ at time $t_{i_2} \in \mathcal{T}, i_2 = 1, \dots, r_2$ corresponding to the model input parameters \mathbf{x} . Sets \mathcal{S} and \mathcal{T} can be thought of as discretised sets of points within continuous spatial and temporal domains \mathbb{S} and \mathbb{T} respectively. That is, \mathcal{S} is a set of coordinate points of interest over a geographical region represented by \mathbb{S} , and \mathcal{T} is a set of time increments (hours or days) over time \mathbb{T} . More generally, if the output arises on a regular grid of higher-order coordinates, the simulator output can be represented by an m -dimensional array or m -order tensor $\mathcal{F}(\mathbf{x}) := f_{i_1 \dots i_m}(\mathbf{x}) : \times_{z=1}^m r_z$, with z th dimension having size r_z , $\mathbf{r}^\top = (r_1, \dots, r_m)$ and $\times_{z=1}^m r_z$ denoting all $r = \prod_{z=1}^m r_z$ combinations of the m indices.

Such output can be modeled using Equation (2.4) via a vectorization operation [34]

$$(2.5) \quad \mathbf{f}(\mathbf{x}) = \text{vec}(\mathcal{F}(\mathbf{x})) = \sum_{\mathbf{i}_r} f_{i_1 \dots i_m}(\mathbf{x}) \boldsymbol{\epsilon}_{i_1 : i_m}^{\mathbf{r}},$$

where $\mathbf{i}_r = \{(i_1, \dots, i_m) : 1 \leq i_z \leq r_z, 1 \leq z \leq m\}$, $\boldsymbol{\epsilon}_{i_1 : i_m}^{\mathbf{r}} = \boldsymbol{\epsilon}_{i_1}^{r_1} \otimes \dots \otimes \boldsymbol{\epsilon}_{i_m}^{r_m}$ and $\boldsymbol{\epsilon}_i^r$ is a r -dimensional unit vector with i th entry 1. However, the coregionalization matrix would have dimension r , possibly making inference and prediction from Equation (2.4) computationally infeasible except in special cases such as output independence and diagonal Σ , as utilized by the PPE.

Alternatively, if an assumption of separability between the output dimensions is reasonable, the coregionalization matrix can be decomposed as

$$(2.6) \quad \Sigma = \Sigma_1 \otimes \dots \otimes \Sigma_m,$$

leading to a *tensor-variate Gaussian process* (TvGP)

$$(2.7) \quad \mathcal{F}(\cdot) \sim \mathcal{TVGP}(\mathcal{M}(\cdot), \kappa(\cdot, \cdot), \Sigma_1, \dots, \Sigma_m),$$

with $\mathcal{M} : \mathbb{R}^p \rightarrow \mathbb{R}^{r_1 \times \dots \times r_m}$ a m -dimensional tensor-variate function. Such a process can be defined via a linear combination of independent univariate standard normal random variables, $\mathbf{u}^\top = (u_1, \dots, u_r)$, $u_l \sim \mathcal{N}(0, 1)$, via

$$\text{vec}(\mathcal{F}(\mathbf{x})) = \text{vec}(\mathcal{M}(\mathbf{x})) + \kappa(\cdot, \cdot)^{\frac{1}{2}} \Sigma^{\frac{1}{2}} \mathbf{u},$$

where Σ has the Kronecker product form from Equation (2.6) and any matrix square root can be used. In an analogy to Equation (2.1), such a process can also be expressed as

$$(2.8) \quad \mathcal{F}(\mathbf{x}) = \mathcal{M}(\mathbf{x}) + \mathcal{E}(\mathbf{x}),$$

where $\mathcal{E} := e_{i_1 \dots i_m}(\mathbf{x}) : \times_{z=1}^m r_z$ and $\mathcal{E} \sim \mathcal{TVGP}(\mathcal{Z}, \kappa(\cdot, \cdot), \Sigma_1, \dots, \Sigma_m)$, with \mathcal{Z} the tensor with all components equal to 0.

A consequence of assuming this prior is that, for a collection of n input vectors $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}$, the $n \times r_1 \times \dots \times r_m$ output tensor $\mathcal{F} = \{\mathcal{F}(\mathbf{x}^{(1)}), \dots, \mathcal{F}(\mathbf{x}^{(n)})\}$ follows a tensor, or multi-linear, normal distribution [34] of dimension $m + 1$

$$\mathcal{F} \sim \mathcal{TN}_{n, r_1, \dots, r_m}(\mathcal{M}, K, \Sigma_1, \dots, \Sigma_m),$$

with $m + 1$ dimensional mean tensor $\mathcal{M} = \{\mathcal{M}(\mathbf{x}^{(1)}), \dots, \mathcal{M}(\mathbf{x}^{(n)})\}$ and K having jk -th entry $\kappa(\mathbf{x}^{(j)}, \mathbf{x}^{(k)})$. Vectorization of \mathcal{F} results in a multivariate normal distribution

$$(2.9) \quad \text{vec}(\mathcal{F}) \sim \mathcal{N}_{nr}(\text{vec}(\mathcal{M}), K \otimes (\otimes_{i=1}^m \Sigma_i)).$$

TvGP regression is defined via the joint distribution of \mathcal{F} and $\tilde{\mathcal{F}} = \{\mathcal{F}(\tilde{\mathbf{x}}^{(1)}), \dots, \mathcal{F}(\tilde{\mathbf{x}}^{(n')})\}$ at new inputs $\tilde{\mathbf{x}}^{(1)}, \dots, \tilde{\mathbf{x}}^{(n')}$, given by

$$\{\mathcal{F}, \tilde{\mathcal{F}}\} \sim \mathcal{TN}_{n+n', r_1, \dots, r_m}(\mathcal{M}_+, K_+, \Sigma_1, \dots, \Sigma_m),$$

with $\mathcal{M}_+ = \{\mathcal{M}, \tilde{\mathcal{M}}\}$, $\tilde{\mathcal{M}} = \{\mathcal{M}(\tilde{\mathbf{x}}^{(1)}), \dots, \mathcal{M}(\tilde{\mathbf{x}}^{(n')})\}$,

$$K_+ = \begin{bmatrix} K & L \\ L^T & \tilde{K} \end{bmatrix},$$

$\tilde{K}_{j'k'} = \kappa(\tilde{\mathbf{x}}^{(j')}, \tilde{\mathbf{x}}^{(k')})$ ($j', k' = 1, \dots, n'$) and $L_{jk'} = \kappa(\mathbf{x}^{(j)}, \tilde{\mathbf{x}}^{(k')})$ ($j = 1, \dots, n; k' = 1, \dots, n'$). Then, conditional on \mathcal{F} we have the posterior predictive distribution

$$\tilde{\mathcal{F}}|\mathcal{F} \sim \mathcal{TN}_{n', r_1, \dots, r_m}(\tilde{\mathcal{M}}^*, \tilde{K}^*, \Sigma_1, \dots, \Sigma_m),$$

with $\tilde{\mathcal{M}}^* = \{\mathcal{M}^*(\tilde{\mathbf{x}}^{(1)}), \dots, \mathcal{M}^*(\tilde{\mathbf{x}}^{(n')})\}$,

$$(2.10) \quad \mathcal{M}^*(\tilde{\mathbf{x}}) = \mathcal{M}(\tilde{\mathbf{x}}) + [\mathcal{F} - \mathcal{M}] \times_1 \kappa(\tilde{\mathbf{x}})^T K^{-1},$$

$\kappa(\tilde{\mathbf{x}}) = (\kappa(\tilde{\mathbf{x}}, \mathbf{x}^{(1)}), \dots, \kappa(\tilde{\mathbf{x}}, \mathbf{x}^{(n)}))^T$, $\tilde{K}_{j'k'}^* = \kappa^*(\tilde{\mathbf{x}}^{(j')}, \tilde{\mathbf{x}}^{(k')})$ and

$$\kappa^*(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}') = \kappa(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}') - \kappa(\tilde{\mathbf{x}})^T K^{-1} \kappa(\tilde{\mathbf{x}}').$$

Here, \times_c is the c -mode tensor-matrix product defined for tensor \mathcal{P} and matrix Λ , with entries $p_{i_1 \dots i_m}$ and λ_{ab} respectively, as

$$(\mathcal{P} \times_c \Lambda)_{i_1 \dots i_{c-1} a i_{c+1} \dots i_m} = \sum_b \lambda_{ab} p_{i_1 \dots i_{c-1} b i_{c+1} \dots i_m}.$$

Hence we have the posterior process

$$\mathcal{F}(\mathbf{x})|\mathcal{F} \sim \mathcal{TVGP}(\mathcal{M}^*(\cdot), \kappa^*(\cdot, \cdot), \Sigma_1, \dots, \Sigma_m).$$

The statistical surrogate model of $\mathbf{f}(\cdot)$ is usually not specified with fixed chosen values for the elements of Σ and θ . If a prior distributional specification is assumed across this collection, the updated posterior distributional form \mathcal{F} is no longer a Gaussian process, and numerical inference methods must be used. Typically, a vectorization of the mean tensor $\mathcal{M}(\cdot)$ is formed as a combination of a linear predictor and unknown parameters (Appendix B). These mean parameters are integrated over with respect to a prior distribution, retaining a posterior predictive Gaussian process. Coregionalization matrices can be parametrized via a kernel function, such as Equation (2.2), or estimated directly from the data under suitable assumptions (see Section 2.6). Any covariance parameters are often estimated as, e.g., posterior modes, and plugged into the posterior predictive density [17, ch. 5].

2.3. Emulation of Dynamic Simulators. A straightforward approach for emulation of spatio-temporal simulators is to treat the $r = r_1 \times r_2$ space-time outputs as a general multivariate collection of random variables and emulate directly using the MvGP from Equation (2.4) [33, 12]. In this case, the indexing of the model output enters only indirectly through specification of the prior variance matrix for the outputs. Such an emulator is simple to implement, however, computationally

expensive for large values of r . It may also be difficult to specify or elicit the required parameters involved in capturing the covariances between all of the outputs.

An alternative approach is to construct a separate emulator for each space-time output, indexed by $i = 1, \dots, r$ [12]:

$$(2.11) \quad f_i(\mathbf{x}) = \mu_i(\mathbf{x}) + e_i(\mathbf{x}),$$

with e_i given independent, zero-mean, Gaussian process priors. As a result, only data for the i th output is utilised when emulating output at spatial-temporal coordinate i . Prior correlation across space-time output is not captured. Additionally, constructing each emulator completely independently, including specification of parameters, is computationally intensive; commonly, r_1 and/or r_2 may be in the order of (tens of) thousands; hence, even if using parallel computation, estimation will be expensive. Even for smaller grid sizes, computation can be reduced by orders of magnitude through the use of a TvGP.

Extensions or variants of these two approaches have proved popular in applications [38, 42, 31, 14]. In the next sections, we place two such methods in the context of tensor-variate Gaussian processes with $m = 2$ dimensional tensor (or matrix) output $\mathcal{F} = f_{i_1 i_2}(\mathbf{x}) : r_1 \times r_2$.

2.4. Outer Product Emulators. An Outer Product Emulator (OPE) for multivariate responses [39] extends the separable emulator (2.4) by assuming a mean function constructed from a further separable structure in the regressors. Such structure can be extended to higher-dimensional outputs, defined over the inputs and all output locations within the TvGP expressed as Equation (2.8). Each component is modeled as

$$f_{i_1 \dots i_m}(\mathbf{x}) = \mathbf{g}_{i_1 \dots i_m}^\top(\mathbf{x})\boldsymbol{\beta} + e_{i_1 \dots i_m}(\mathbf{x}), \quad i_z = 1, \dots, r_z; \quad z = 1, \dots, m,$$

where $\mathbf{g}_{i_1 \dots i_m}^\top(\mathbf{x}) = \mathbf{g}_0^\top(\mathbf{x}) \otimes (\otimes_{z=1}^m \mathbf{g}_z^\top(\boldsymbol{\xi}_{z i_z}))$, $\boldsymbol{\beta} = \boldsymbol{\beta}_0 \otimes (\otimes_{z=1}^m \boldsymbol{\beta}_z)$, with $\mathbf{g}_0(\cdot)$ and $\boldsymbol{\beta}_0$ being v_0 -vectors of regressors and parameters related to the input variables, and $\mathbf{g}_z(\cdot)$ and $\boldsymbol{\beta}_z$ v_z -vectors corresponding to the output locations in dimension z . Hence, the TvGP has mean tensor

$$(2.12) \quad \mathcal{M}(\mathbf{x}) = \mathbf{g}_{i_1 \dots i_m}^\top(\mathbf{x})\boldsymbol{\beta} : \times_{z=1}^m r_z.$$

For the $m = 2$ dimensional spatio-temporal case, we define $\boldsymbol{\xi}_{1 i_1} = s_{i_1} \in \mathcal{S}$ as the spatial locations and temporal locations $\boldsymbol{\xi}_{2 i_2} = t_{i_2} \in \mathcal{T}$ and

$$\begin{aligned} \mathcal{M}(\mathbf{x}) &= \begin{bmatrix} \mathbf{g}_{11}^\top(\mathbf{x})\boldsymbol{\beta} & \cdots & \mathbf{g}_{1 r_2}^\top(\mathbf{x})\boldsymbol{\beta} \\ \vdots & \ddots & \vdots \\ \mathbf{g}_{r_1 1}^\top(\mathbf{x})\boldsymbol{\beta} & \cdots & \mathbf{g}_{r_1 r_2}^\top(\mathbf{x})\boldsymbol{\beta} \end{bmatrix}, \\ &= G(\mathbf{x}) [I_{r_2} \otimes \boldsymbol{\beta}] \end{aligned}$$

where

$$G(\mathbf{x}) = \begin{bmatrix} \mathbf{g}_{11}^\top(\mathbf{x}) & \cdots & \mathbf{g}_{1 r_2}^\top(\mathbf{x}) \\ \vdots & \ddots & \vdots \\ \mathbf{g}_{r_1 1}^\top(\mathbf{x}) & \cdots & \mathbf{g}_{r_1 r_2}^\top(\mathbf{x}) \end{bmatrix},$$

and I_{r_2} is an $r_2 \times r_2$ identity matrix.

The TvGP is completed by specification of coregionalization matrix $\Sigma = \otimes_{z=1}^m \Sigma_z$ for residual tensor \mathcal{E} . For emulators with spatio-temporal output,

$$\Sigma = \Sigma_1 \otimes \Sigma_2,$$

with Σ_1 an $r_1 \times r_1$ spatial coregionalization matrix and Σ_2 an $r_2 \times r_2$ temporal coregionalization matrix. Advantage can be taken of naturally defined metric spaces on the output locations to

borrow strength across different space-time coordinates via suitable prior specifications. Here, we assume $\Sigma = \sigma^2 W_1 \otimes W_2$. Hence $\Sigma_1 = W_1$, with $r_1 \times r_1$ matrix W_1 having entries defined via a kernel specifying response correlation on the spatial output locations, and $\Sigma_2 = \sigma^2 W_2$, where $r_2 \times r_2$ matrix W_2 is similarly defined via a kernel for response correlation on the temporal output locations. The single scale parameter is σ^2 . Efficient computational methods are available for the OPE with matrix-variate output as appropriate for spatio-temporal simulators; see [4] and Appendix B.1, which demonstrates that a vectorization of the tensor-variate OPE maintains the usual form of a vector-variate OPE.

2.5. Parallel Partial Emulators. A Parallel Partial Emulator (PPE), introduced in [18], involves fitting separate, shared-characteristic emulators at each output grid point:

$$(2.13) \quad f_{i_1 \dots i_m}(\mathbf{x}) = \mathbf{g}_0^\top(\mathbf{x}) \boldsymbol{\beta}_{i_1 \dots i_m} + e_{i_1 \dots i_m}(\mathbf{x}).$$

The emulators have a common v_0 -vector of basis functions $\mathbf{g}_0(\cdot)$, but distinct v_0 -vectors of regression coefficients $\boldsymbol{\beta}_{i_1 \dots i_m}$. Emulators at different output locations are assumed conditionally independent, a priori, and hence $\Sigma = \text{diag}(\sigma_1^2, \dots, \sigma_r^2)$. The PPE can be expressed as a TvGP with a single output dimension of size $r = \prod_{z=1}^m r_z$, that combines all output indices. This is equivalent to MvGP (2.4). The emulator has mean vector

$$(2.14) \quad \mathcal{M}(\mathbf{x}) = \left[I_r \otimes \mathbf{g}_0^\top(\mathbf{x}) \right] \text{vec}(\mathcal{B}),$$

with $\mathcal{B} = \beta_{i_1 \dots i_m, j} : [\times_{z=1}^m r_z] \times v_0$. Here \mathcal{B} is a tensor representation of the output-specific $(i_1 \dots i_m)$ coefficients of the common v_0 regressors and $\beta_{i_1 \dots i_m, j}$ is the j th element of $\boldsymbol{\beta}_{i_1 \dots i_m}$.

The shared structure of the linear predictor and kernel function, including common parameters $\boldsymbol{\theta}$ results in the PPE inheriting properties of the simulator, including smoothness [18]. Completely independent emulators for each space-time coordinate would not have this property. Mature code for estimation and inference using the PPE is available in the **RobustGaSP** package [20] for R [36].

2.6. Priors and estimation. Connections between the OPE and PPE can be established via the choice of priors [39, 12, 18], i.e., for $\boldsymbol{\beta}$ and Σ_z . Here, for the OPE we use a conjugate normal-inverse gamma joint prior for $\boldsymbol{\beta}, \sigma^2$ [39], leading to a marginal multivariate t -distribution prior for $\boldsymbol{\beta}$. The specification of a common prior variance for the OPE can lead to inferior uncertainty quantification compared to the PPE, see Sections 3.3 and 4.3.

In the mean structure, for both emulators, we specify input regressors $\mathbf{g}_0^\top = (1, \mathbf{x}^\top)$. For the OPE we also specify temporal regressors $\mathbf{g}_2^\top = (1, t)$. It is important to note that this inclusion of output locations in the OPE prior mean will result in differences in the posterior predictive means for the OPE and PPE; see Appendix B.2.

The input correlation for both emulators was specified as Gaussian from Equation (2.3). For the OPE, the Gaussian correlation function was also used to parameterize the temporal correlations, with

$$(\Sigma_2)_{i_2 i'_2} = \sigma^2 \exp\left(-\{t_{i_2} - t_{i'_2}\}^2 / \theta_2\right),$$

and hyperparameter $\theta_2 > 0$. The choice of spatial regressors and correlation function for the OPE varies between our two examples.

The OPE is completed via choice of parameters for the normal-inverse gamma prior discussed above; we use $\sigma^{-2} \sim \text{Gamma}(1, 1)$ and $\boldsymbol{\beta} | \sigma^2 \sim N(\mathbf{0}, \sigma^2 I_{v_0 v_1 v_2})$. Hence, the marginal prior distribution for $\boldsymbol{\beta}$ is a scaled t -distribution with two degrees of freedom, and enforces shrinkage towards zero for the posterior of the regression parameters. All correlation parameters were estimated via maximum likelihood, which corresponds to minimizing the Mahalanobis distance, using the L-BFGS-B algorithm [8].

For the PPE, vague prior beliefs are assumed for $\beta_{i_1 i_2}$, leading to posterior mean and variance estimators for $\beta_{i_1 i_2}$ equivalent to generalized least squares estimators [25]. We obtain point estimates for σ_i^2 and common correlation parameters θ via maximum likelihood [1, 27], with the latter obtained numerically.

The OPE and PPE employ different paradigms for the specification and estimation of the coregionalization matrices $\Sigma_1, \dots, \Sigma_m$. The OPE parameterizes these matrices via a kernel function, such as Equation (2.3), whereas for the PPE, they are estimated directly from the data, without assuming a parametric form (albeit assuming prior independence across output locations). Recent research for both these approaches has focused on fast computation and effective uncertainty quantification via latent factor modeling [21, 19, 32]. The synthesis of such methods with the TvGPs proposed in our work is an area for future research.

Two different computer experiments were designed and performed to generate training data, with $n = 20$ (\mathcal{X}_{20}) and $n = 50$ (\mathcal{X}_{50}). These sizes of design were chosen to mimic the common use case of only limited data availability from computationally expensive simulators. Both designs were found using the MaxPro [28] space-filling criterion, with input dimensions each scaled to $[-1, 1]$. All statistical modeling also used this scaling, which was also applied to spatial and temporal locations. A further MaxPro design, \mathcal{X}_D of size $n' = 150$, was constructed and run for diagnostic purposes.

3. Application: Dynamic Influenza Simulator.

3.1. The Simulator. Our first application is to the compartmental SEIR simulator for influenza introduced in Section 1. The spatial patches [44] represent population distributions among 15 fictional regions. The population in each patch moves through the four states (S , E , I and R) over time according to a system of deterministic differential equations, governed by a set of rate parameters which are consistent across all patches. Previous work on emulating SEIR simulators has not included such a spatial component [13].

Interactions between infection levels in neighboring patches are represented by a non-symmetric (commuter) matrix, through which movement between the different patches is modeled [41], with row a column b representing the flow of people from patch a to patch b . Two daily timesteps are assumed, simulating travel from a “home” patch to a “work” patch and back again. The susceptible population in patch a can become infected if they mix with (i) infected individuals from patch a ; (ii) infected individuals in patch a who have moved from patch b ; or (iii) infected individuals within patch b if they themselves commute to patch b .

Here, spatio-temporal emulators are built for the number of infected individuals in each patch over time. The simulator is initialized with 100 infected individuals in a single fixed patch with the remaining population across all patches being susceptible. The input parameters to the simulator are rates $x_1 = \beta \in [1, 2]$, $x_2 = \alpha \in [0.25, 1]$ and $x_3 = \gamma \in [0.25, 1]$, which are constant across space and time and effectively control the modeled transmission rates from S to E , E to I and I to R , respectively. Initial conditions for the simulator and the ranges for these inputs were chosen by the epidemiological modelers at UKHSA.

3.2. Emulation. Tensor-variate GP emulators were constructed for this simulator, under each of the OPE (Section 2.4) and PPE assumptions (Section 2.5). The simulator output at each space-time location, $y_{i_1 i_2}(\mathbf{x})$, was shifted and log transformed before emulation; that is, $f_{i_1 i_2}(\mathbf{x}) = \log(y_{i_1 i_2}(\mathbf{x}) + 1)$. Each simulator run produced numbers of infected individuals across $r_1 = 15$ spatial patches and $r_2 = 150$ discrete time steps.

The OPE and PPE were constructed using the priors described in Section 2.6. The fixed initial conditions and commuter matrix led, for all combinations of input variables, to the same temporal order of patches obtaining their respective maximum number of infections. This ordering was used to define a new spatial coordinate for each patch, \tilde{s}_{i_1} , taking values in $\{1, \dots, 15\}$, which are common across all simulator runs. Hence these quantities can be used to define Σ_1 for the OPE.

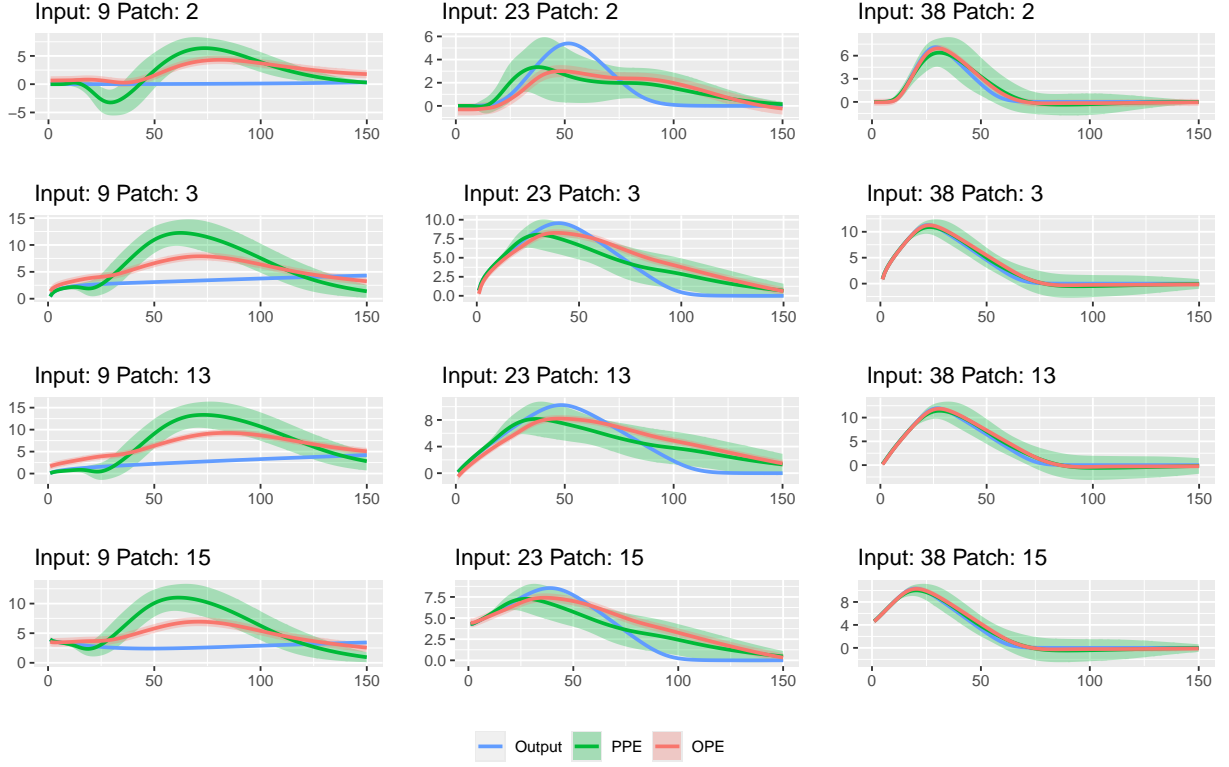


Figure 1: Emulator mean predictions (solid lines) and 95% credible intervals (shaded area) for the influenza simulator for three test input points (columns) and four patches (rows). The OPE (red) and PPE (green) were built using $n = 20$ training points. In each case, true simulator output is given as a blue line for comparison. Predictions and responses are on the log scale.

Table 1: Diagnostic input parameters for the influenza simulator corresponding to the points in Figures 1 and 2, scaled to $[-1, 1]$.

Input	α	β	γ
9	0.19	-0.92	0.99
23	0.68	-0.73	0.20
38	0.67	-0.54	-0.78

This matrix was parameterized using a Gaussian correlation function and therefore for the OPE

$$(\Sigma_1)_{i_1 i'_1} = \exp\left(-\{\tilde{s}_{i_1} - \tilde{s}_{i'_1}\}^2 / \theta_1\right),$$

with hyperparameter $\theta_1 > 0$. For the prior mean structure, we use $\mathbf{g}_1^\top(\hat{s}_{i_1}) = (1, \hat{s}_{i_1})$, where \hat{s}_{i_1} is a scaled patch size.

3.3. Results. Figures 1 ($n = 20$) and 2 ($n = 50$) give emulator predictions for four arbitrarily selected patches and three points from the diagnostic set \mathcal{X}_D . The values of the input parameters for these three diagnostic points are given in Table 1.

Predictions for input 9 were notably poor for both OPE and PPE emulators, regardless of training design size (left-hand columns of Figure 1 and Figure 2). Although input 9 is actually closer to a design point than either inputs 23 or 38, this point is near the edge of the input space (Table 1 and Figure A.1), limiting the ability of the emulators to borrow information from many

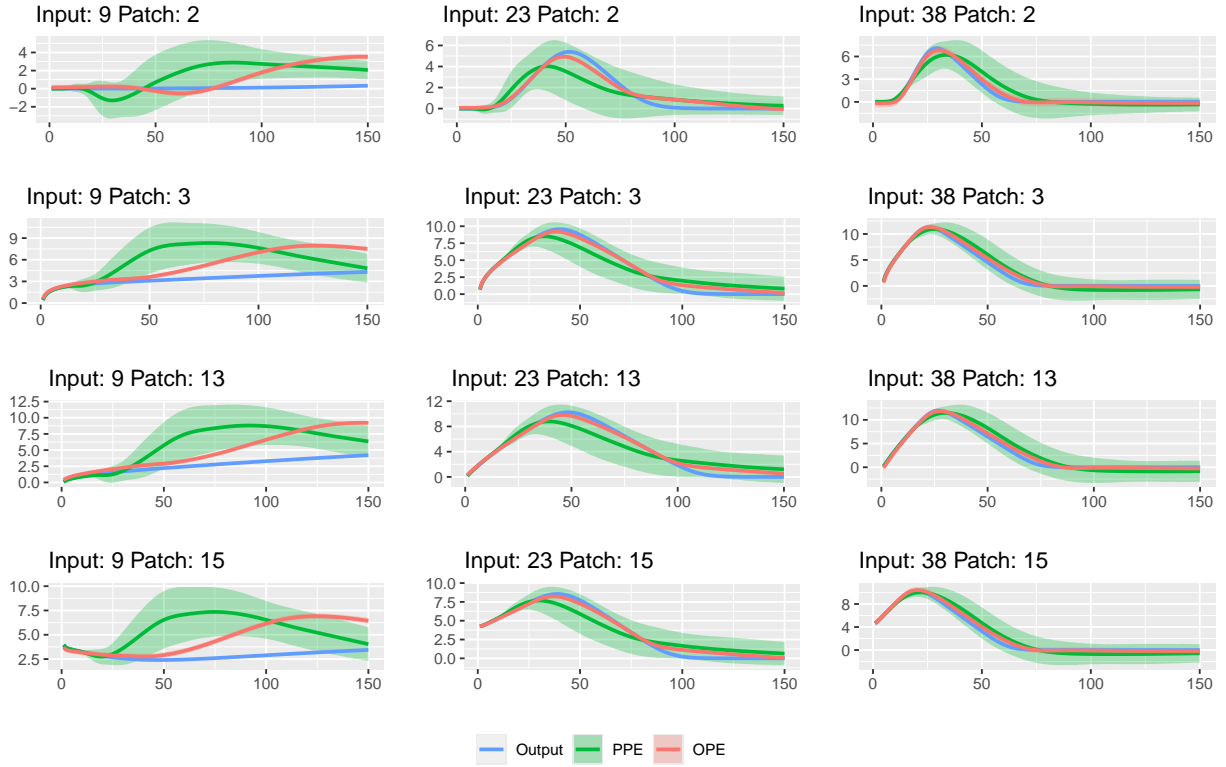


Figure 2: Emulator mean predictions (solid lines) and 95% credible intervals (shaded area) for the influenza simulator for three test input points (columns) and four patches (rows). The OPE (red) and PPE (green) were built using $n = 50$ training points. In each case, true simulator output is given as a blue line for comparison. Predictions and responses are on the log scale.

nearby points. Additionally, and importantly, the specific parameter values associated with input 9 also resulted in minimal disease dynamics, with almost no infections in any patch. This type of scenario was rare in the training and test designs and hence poses a challenge for the emulators to capture accurately.

For a more holistic comparison of the emulators, Figure 3 graphically assesses prediction for 10^4 input-space-time points (combinations of $\mathbf{x}^{(i)}$, \tilde{s}_{i_1} or s_{i_1} and t_{i_2}), randomly selected from the diagnostic set \mathcal{X}_D . Table 2 gives numerical summaries of the predictive performance of each emulator (with definitions of the summaries in Appendix A). Unsurprisingly, both emulators are improved by increasing the training design size from $n = 20$ to $n = 50$. The differing structure of the models is also apparent in the results; for example, the predictive variance for the OPE is constant across space-time, whereas each location has a potentially different variance for the PPE. This feature leads to more appropriate uncertainty quantification for the PPE, reflected in the the mean absolute squared prediction error (MASPE) and mean generalized entropy score (MGES) values in Table 2. However, the OPE does appear more accurate (lower root mean squared prediction error (RMSPE)) than the PPE for both $n = 20$ and $n = 50$, possibly due to the spatial-temporal structure incorporated through both the choice of regressors and space/time correlation matrices.

Clearly, an advantage of the OPE is the direct and explicit incorporation of the spatio-temporal structure and exploitation of any continuity in these dimensions. This feature was somewhat negated in the influenza example by the limited, and discrete, nature of the spatial information. In the next section, we demonstrate the benefits of the OPE of a metric space in both the space and time dimensions.

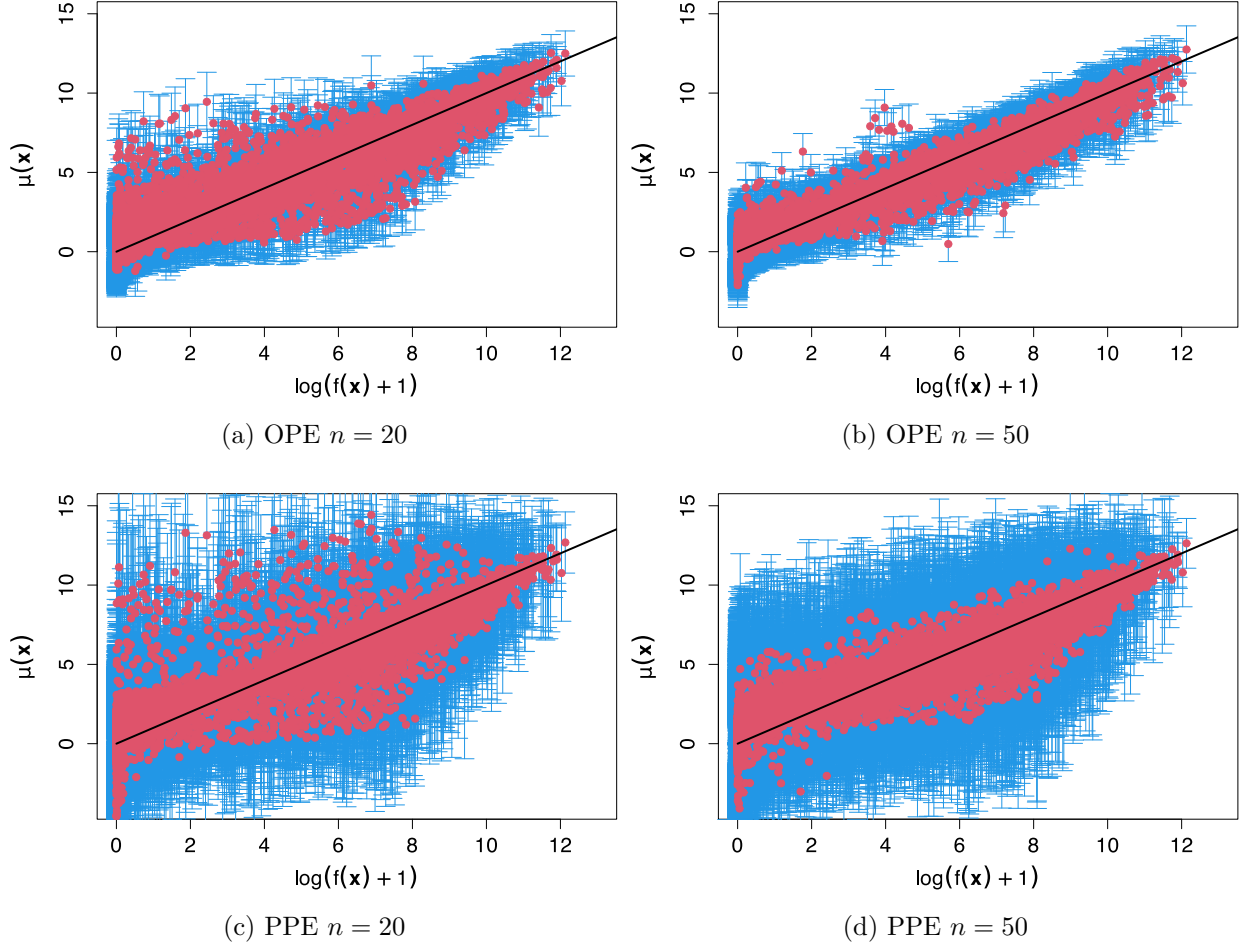


Figure 3: Emulator expectation ± 3 standard deviations against influenza simulator output for 10^4 points (input, space, time) from the set \mathcal{X}_D of $n' = 150$ diagnostic runs for OPE and PPE with $n = 20$ and $n = 50$ training points. Predictions and responses are on the log scale.

Table 2: The mean absolute squared prediction error (MASPE)¹, root mean squared prediction error (RMSPE)² and mean generalized entropy score (MGES)³ for each of the four approximations and two design sizes discussed in Sections 3 and 4. The performance measures are calculated on the test set \mathcal{X}_D and on the log scale. See Appendix A for definitions.

Simulator	Diagnostic	$n = 20$		$n = 50$	
		OPE	PPE	OPE	PPE
Influenza	MASPE	2.758	0.736	2.142	0.664
	RMSPE	1.107	1.424	0.632	1.080
	MGES ($\times 10^3$)	-4586	-270	-2692	-37
Environmental	MASPE	0.818	0.890	0.724	0.823
	RMSPE	0.112	0.133	0.069	0.149
	MGES ($\times 10^3$)	814	1260	1047	1885

¹ MASPE should be broadly close to 1. ² RMSPE is a smaller-the-better quantity.

³ MGES is a larger-the-better quantity (including corresponding sign).

4. Application: Synthetic Environmental Simulator.

4.1. The Simulator. As a second example, consider an environmental simulator modeling a pollutant spill caused by a chemical accident [5]. A mass $x_1 \in [7, 13]$ of pollutant is spilled at each of two locations in a long, narrow channel, denoted by the space-time vectors $(0, 0)$ and (x_2, x_3) . The first spill is fixed; however, the time and spatial location of the second spill are inputs. The location of the second spill is almost the entire domain, $x_2 \in [0.01, 3]$, but the time of the second spill is quite restrictive, $x_3 \in [30.01, 30.295]$. A fourth input variable, $x_4 \in [0.02, 0.12]$, is the constant diffusion rate in the channel.

The simulator output, log pollutant concentration at spatial location $s_{i_1} \in [0, 3]$ and time $t_{i_2} \in [0, 60]$ is defined as the following closed-form function of $\mathbf{x} = (x_1, \dots, x_4)^\top$:

$$f_{i_1 i_2}(\mathbf{x}) = \log \left\{ \sqrt{4\pi} C(\mathbf{x}; s_{i_1}, t_{i_2}) + 1 \right\} \quad i_1 = 1, \dots, r_1; i_2 = 1, \dots, r_2, \quad (4.1)$$

$$C(\mathbf{x}; s, t) = \frac{x_1}{\sqrt{4\pi x_4 t}} \exp \left(\frac{-s^2}{4x_4 t} \right) + \frac{x_1}{\sqrt{4\pi x_4 (t - x_3)}} \exp \left(\frac{-(s - x_2)^2}{4x_4 (t - x_3)} \right) \mathbb{I}(x_3 > t),$$

where \mathbb{I} is the indicator function. Figure 4a shows the simulator output as a function of s and t for the values $x_1 = 10, x_2 = 1.505, x_3 = 30.1525$ and $x_4 = 0.07$.

Note that the time and location of the first release are fixed and the time of the second release varies slightly around the halfway point of the time interval. Therefore, for $t \in (0, 30)$, approximately, the simulator output only depends on x_1 , the mass of pollutant spilled at each location, and x_4 , the diffusion rate. Hence, emulation of the output in this time region is straightforward. However, for $t \in (30, 60)$, approximately, the output is very sensitive to the other inputs, making emulation more difficult, see Figure 4b. This is especially true (i) around the second release at $t \approx 30$; and (ii) if a good correlation structure is not assumed for the output space.

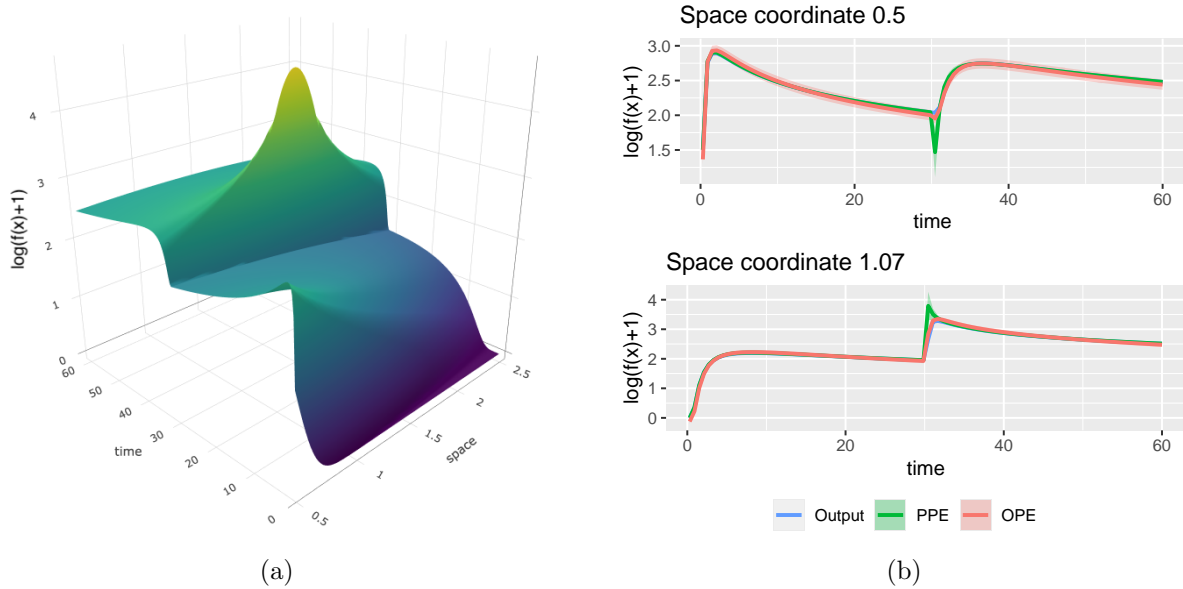


Figure 4: Output for the environmental simulator (4.1) for the input values $x_1 = 10, x_3 = 30.1525, x_2 = 1.505$ and $x_4 = 0.07$. (a) As a function of space and time. (b) Temporal slices at spatial locations $s = 0.5$ and $s = 1.07$ with OPE and PPE mean predictions ± 3 standard deviations from $n = 50$ training points. Predictions and responses are on the log scale.

4.2. Emulation. Once again, tensor-variate GP emulators were constructed under each of the OPE (Section 2.4) and PPE (Section 2.5) assumptions using designs and prior distributions from Section 2.6. For the OPE, the prior mean and correlation structure directly used spatial locations, with $\mathbf{g}_1^\top(s_{i_1}) = (1, s_{i_1})$ and

$$(\Sigma_1)_{i_1 i'_1} = \exp\left(-\{s_{i_1} - s_{i'_1}\}^2/\theta_1\right),$$

with hyperparameter $\theta_1 > 0$. Each simulator run produced concentrations for $r_1 = 15$ locations and $r_2 = 100$ times, defined by the discrete sets $s_{i_1} \in \mathcal{S} = \{0.50, 0.64, \dots, 2.50\}$ and $t_{i_2} \in \mathcal{T} = \{0.3, 0.9, \dots, 60\}$ [5].

4.3. Results. Figure 4b shows examples of two sets of temporal predictions from the OPE and PPE for different spatial locations and one set of input parameters. The key feature revealed by these plots is the difficulty exhibited by PPE in predicting around the time of the second release, with a spike in variance and either under- or over-prediction in the mean. These results are fairly representative of the overall, qualitative, comparison of the two emulators, see Figure A.2 for further examples.

As in Section 3, Figure 5 provides an assessment of predictive performance for 10^4 input-space-time points (combinations of $\mathbf{x}^{(i)}$, s_{i_1} and t_{i_2}), randomly selected from the diagnostic set \mathcal{X}_D ; numerical summaries (MASPE, RMSPE, MGES) are again provided in Table 2. The difficulty of PPE to predict at the second release is evident in the much larger (than OPE) prediction uncertainty for larger simulator outputs, especially around $f_{i_1 i_2}(\mathbf{x}) = 2$. OPE performs better for these difficult predictions, albeit with higher variance for small values of $f_{i_1 i_2}(\mathbf{x})$ due to the constant space-time variance assumption. This assumption also leads to PPE having larger (better) MGES values in Table 2, influenced by more precise predictions in these regions. However, the OPE has substantially lower RSMPE values, especially for $n = 50$. Interestingly, increasing the run size from $n = 20$ to $n = 50$ does little to help with the poor PPE predictions around $f_{i_1 i_2}(\mathbf{x}) = 2$, although it does reduce prediction variance even more for small $f_{i_1 i_2}(\mathbf{x})$. For OPE, increasing the design size leads to an overall decrease in prediction variance.

The combination of a metric space in both the space and time locations [39], combined with spatio-temporal structure in both the regressors and correlation function, provides OPE with an advantage in this example. The concentration output at a given spatial location near the time of the second release can be very variable, depending on the inputs (especially the location of the second release). Hence the PPE near that time point can have high uncertainty, as it does not have the same ability to borrow strength across the space-time locations.

5. Conclusions. Tensor-variate Gaussian processes are a powerful tool for emulation of multi-output simulators, which generalize previous approaches. Applications of such simulators are common in areas such as biomedicine [43] and engineering [24], in addition to spatio-temporal modeling. There are also synergies with multi-task learning using Gaussian processes [6].

The important special cases of OPE and PPE have been compared across two case studies, using a practically important simulator from public health and a synthetic environmental simulator. PPE has modeling advantages when the simulator outputs are unstructured, or the structure adds little to the understanding of the system, particularly if larger computer experiments can be performed. When an advantage can be taken of correlation in the output dimensions, OPE can provide more accurate predictions, especially in cases where output structure can compensate for high variation with respect to the simulator inputs. This higher accuracy does come at a higher computational cost, especially for larger numbers of output locations (for the influenza simulator, emulation estimation and prediction for the OPE took 2.77 minutes as opposed to 1.43 seconds for the PPE, as implemented in R on a MacBook Pro with Apple M1 Pro (16 GB) processor). However, for many applications, computational expense will be dominated by the computer experiment on the simulator, rather than the cost of fitting the emulator.

An obvious extension of this work would employ TvGP emulators that combine features from

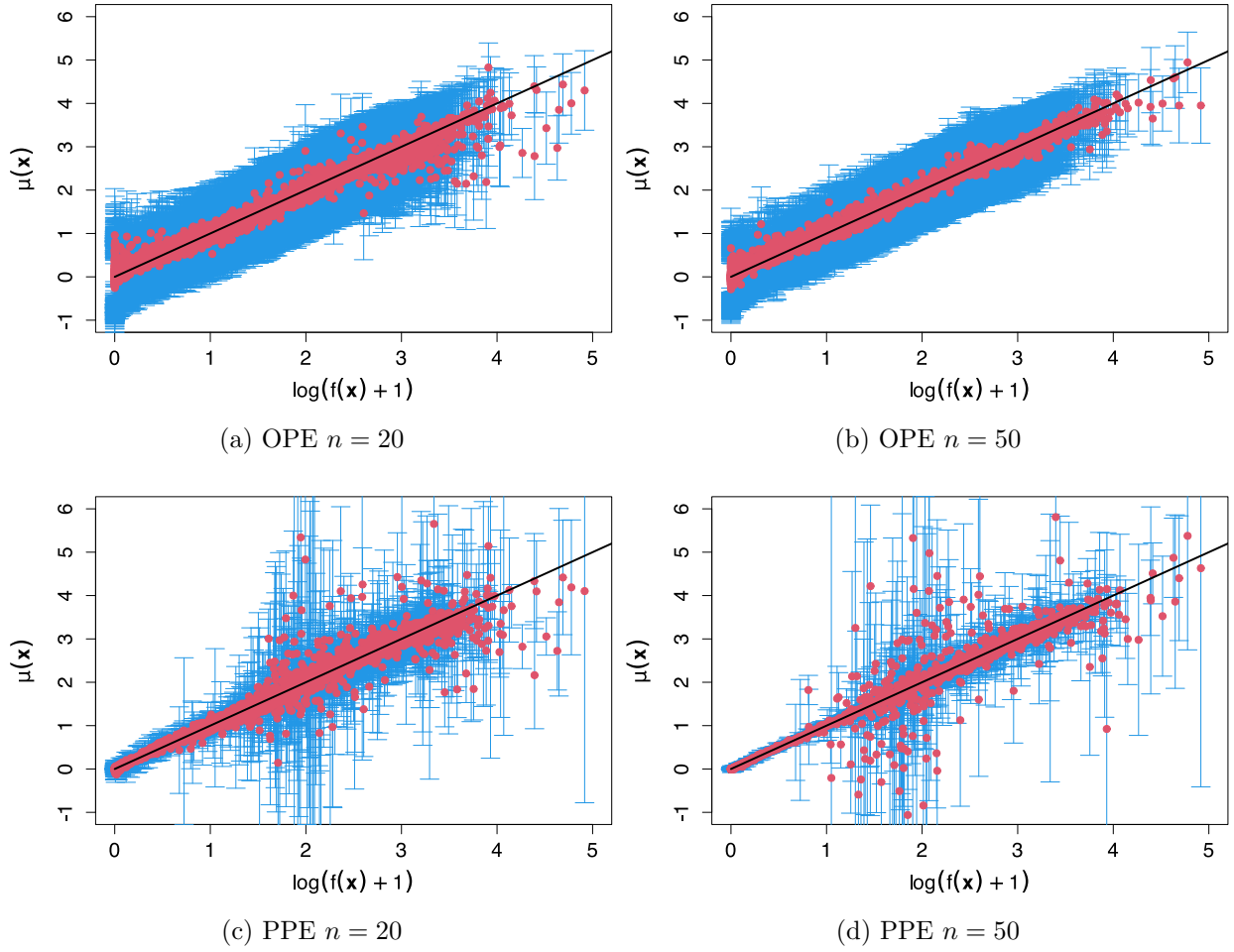


Figure 5: Emulator expectation ± 3 standard deviations against environmental simulator output value for 10^4 points (input, space, time) from the set \mathcal{X}_D of $n' = 150$ diagnostics runs for OPE and PPE with $n = 20$ and $n = 50$ training points. Predictions and responses are on the log scale.

both the common OPE and PPE approaches. For example, on the dynamic influenza example we also fitted a variant of the OPE which relaxed the assumption of constant prior variance across the spatial patches and instead assumed prior independence of the response across these patches. That is, $W_1 = \text{diag}\{\sigma_{11}, \dots, \sigma_{1r_1}\}$. This emulator has r_1 variance parameters, contrasting to a single variance parameter for the standard OPE and $r_1 \times r_2$ for the PPE. Parameters were once again estimated using maximum likelihood. This model obtained very similar accuracy as the standard OPE but with improved uncertainty quantification (e.g., for $n = 50$, MASPE of 0.794 and MGES of -467). However, it still lagged behind the PPE in uncertainty quantification properties, cf. Table 2.

Further work could address the additional computation complexity of the TvGP regression, particularly the OPE, through kernel choice for the coregionalization matrices (e.g., tapered kernels [29]) or include heteroskedastic errors across output dimensions. Variable selection could be considered for general TvGP regression and for the OPE in particular, including the choice of regressors in the output locations. TvGP regression could also be applied to provide prediction for output locations that differ from those of the simulator, provided a continuous output domain can be defined. Various TvGP emulators could also be assessed and compared for higher-dimensional problems; for example, including additional compartments (e.g., susceptible or exposed) in the influenza example.

Acknowledgments. We are grateful to CrystalCast project members for their invaluable comments and discussion. Particular thanks are due to Professor Veronica Bowman (Paebbl), Dr Hannah Williams (Defence Science and Technology Laboratory, UK), and Dr Thomas Finnie (UK Health Security Agency) for providing the influenza simulator and associated discussions.

References.

- [1] Y. Andrianakis and P. G. Challenor. The effect of the nugget on Gaussian process emulators of computer models. *Computational Statistics and Data Analysis*, 56:4215–4228, 2012.
- [2] T. S. Bastos and A. O’Hagan. Diagnostics for Gaussian process emulators. *Technometrics*, 51:425–438, 2008.
- [3] J. M. Bayarri, J. O. Berger, R. Paulo, J. Sacks, J. A. Cafeo, J. Cavendish, C. Lin, and J. Tu. A framework for validation of computer models. *Technometrics*, 49:138–154, 2007.
- [4] I. Bilonis, N. Zabaras, B. A. Konomi, and G. Lin. Multi-output separable gaussian process: Towards an efficient, fully bayesian paradigm for uncertainty quantification. *Journal of Computational Physics*, 241:212–239, 2013.
- [5] N. Bliznyuk, D. Ruppert, C. Shoemaker, R. Regis, S. Wild, and P. Mugunthan. Bayesian calibration and uncertainty analysis for computationally expensive models using optimization and radial basis function approximation. *Journal of Computational and Graphical Statistics*, 17:270–294, 2008.
- [6] E. V. Bonilla, K. M. A. Chai, and C. K. I. Williams. Multi-task Gaussian process prediction. In *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007.
- [7] V. E. Bowman and D. C. Woods. Emulation of multivariate simulators using thin-plate splines with application to atmospheric dispersion. *SIAM/ASA Journal of Uncertainty Quantification*, 4:1323–1344, 2016.
- [8] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16:1190–1208, 1995.
- [9] M. Calder, C. Craig, D. Culley, R. de Cani, C. A. Donnelly, R. Douglas, B. Edmonds, J. Gascoigne, N. Gilbert, C. Hargrove, D. Hinds, D. C. Lane, D. Mitchell, G. Pavay, D. Robertson, B. Rosewell, S. Sherwin, M. Walport, and A. Wilson. Computational modelling for decision-making: where, why, what and how. *Royal Society Open Science*, 5:172096, 2018.
- [10] A. Campbell and P. Pietro Liò. tvGP-VAE: Tensor-variate Gaussian process prior variational autoencoder. *arXiv:2006.04788*, 2020.
- [11] W. Chu and Z. Ghahramani. Probabilistic models for incomplete multi-dimensional arrays. In D. van Dyk and M. Welling, editors, *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, pages 89–96, Clearwater Beach, Florida, 2009. Proceedings of Machine Learning Research.
- [12] S. Conti and A. O’Hagan. Bayesian emulation of complex multi-output and dynamic computer models. *Journal of Statistical Planning and Inference*, 140:640–651, 2010.
- [13] M. Farah, P. Birrell, S. Conti, and D. De Angelis. Bayesian emulation and calibration of a dynamic epidemic model for A/H1N1 influenza. *Journal of the American Statistical Association*, 109:1398–1411, 2014.
- [14] Y. Gao and E. B. Pitman. Parallel partial emulation in applications. *International Journal for Uncertainty Quantification*, 14:1–15, 2024.
- [15] T. Gneiting and A. E. Raftery. Strictly proper scoring rules, prediction and estimation. *Journal of the American Statistical Association*, 102:359–378, 2007.
- [16] M. Goldstein and J. C. Rougier. Probabilistic formulations for transferring inferences from mathematical models to physical systems. *SIAM Journal on Scientific Computing*, 26:467–487, 2004.
- [17] Robert B. Gramacy. *Surrogates: Gaussian Process Modeling, Design and Optimization for the Applied Sciences*. Chapman Hall/CRC, Boca Raton, Florida, 2020. <http://bobby.gramacy>.

- com/surrogates/.
- [18] M. Gu and J. O. Berger. Parallel partial Gaussian process emulation for computer models with massive output. *Annals of Applied Statistics*, 10:1317–1347, 2016.
- [19] M. Gu and H. Li. Gaussian orthogonal latent factor processes for large incomplete matrices of correlated data. *Bayesian Analysis*, 17:1219–1244, 2022.
- [20] M. Gu, J. Palomo, and J. O. Berger. RobustGaSP: robust Gaussian stochastic process emulation in R. *The R Journal*, 11:112–136, 2019.
- [21] M. Gu and W. Shen. Generalized probabilistic principal component analysis of correlated data. *Journal of Machine Learning Research*, 21:1–41, 2020.
- [22] H. W. Hethcote. The mathematics of infectious diseases. *SIAM Review*, 42:599–653, 2000.
- [23] D. Higdon, J. Gattiker, B. Williams, and M. Rightley. Computer model calibration using high-dimensional output. *Journal of the American Statistical Association*, 103:570–583, 2008.
- [24] Toal. D. J. J. Applications of multi-fidelity multi-output Kriging to engineering design optimization. *Structural and Multidisciplinary Optimization*, 66:125, 2023.
- [25] S. E. Jackson. *Design of Physical System Experiments Using Bayes Linear Emulation and History Matching Methodology with Application to Arabidopsis Thaliana*. PhD thesis, Durham University, 2018.
- [26] S. E. Jackson and I. Vernon. Efficient emulation of computer models utilising multiple known boundaries of differing dimension. *Bayesian Analysis*, 18:165–191, 2023.
- [27] S. E. Jackson, I. Vernon, J. Liu, and K. Lindsey. Understanding hormonal crosstalk in arabidopsis root development via emulation and history matching. *Statistical Approaches in Genetics and Molecular Biology*, 19, 2020.
- [28] V. R. Joseph, E. Gul, and S. Ba. Maximum projection designs for computer experiments. *Biometrika*, 102:371–380, 2015.
- [29] C. G. Kaufman, D. Bingham, S. Habib, K. Heitmann, and J. A. Frieman. Efficient emulators of computer experiments using compactly supported correlation functions, with an application to cosmology. *The Annals of Applied Statistics*, 5:2470 – 2492, 2011.
- [30] M. C. Kennedy and A. O’Hagan. Bayesian calibration of computer models (with discussion). *Journal of the Royal Statistical Society B*, 63:425–464, 2001.
- [31] A. P. Kyprioti, C. Irwin, A. A. Taflanidis, N. C. Nadal-Caraballo, M. C. Yawn, and L. A. Aucoin. Spatio-temporal storm surge emulation using gaussian process techniques. *Coastal Engineering*, 180:104231, 2023.
- [32] Y. Lin, X. Liu, P. Segall, and Gu. M. Fast data inversion for high-dimensional dynamical systems from noisy measurements. *arXiv:2501.01324*., 2025.
- [33] A. O’Hagan, M. C. Kennedy, and J. E. Oakley. Uncertainty analysis and other inference tools for complex computer codes. In J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, editors, *Bayesian Statistics 6*, pages 503–524. Oxford University Press, 1999.
- [34] M. Ohlson, M. Rauf Ahmad, and D. von Rosen. The multilinear normal distribution: introduction and some basic properties. *Journal of Multivariate Analysis*, 113:37–47, 2013.
- [35] A. M. Overstall and D. C. Woods. Multivariate emulation of computer simulators: model selection and diagnostics with application to a humanitarian relief model. *Journal of the Royal Statistical Society C*, 65, 2016.
- [36] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2025.
- [37] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, Massachusetts, 2006.
- [38] J. Rougier, S. Guillas, A. Maute, and A. D. Richmond. Expert knowledge and multivariate emulation: The thermosphere-ionosphere electrodynamics general circulation model (TIE-GCM). *Technometrics*, 51:414–424, 2009.
- [39] J. C. Rougier. Efficient emulators for multivariate deterministic functions. *Journal of Com-*

- putational and Graphical Statistics, 17:827–843, 2008.
- [40] T. J. Santner, B. J. Williams, and W. I. Notz. *The Design and Analysis of Computer Experiments*. Springer, New York, 2nd edition, 2018.
- [41] L. Sattenspiel and K. Dietz. A structured epidemic model incorporating geographic mobility among regions. *Mathematical Biosciences*, 128:71–91, 1995.
- [42] E. T. Spiller, M. J. Bayarri, J. O. Berger, E. S. Calder, A. K. Patra, E. B. Pitman, and R. L. Wolpert. Automating emulator construction for geophysical hazard maps. *SIAM/ASA Journal of Uncertainty Quantification*, 2:126–152, 2014.
- [43] P. Stolfi and F. Castiglione. Emulation of dynamic multi-output simulator of risk of type-2 diabetes. In *2020 IEEE International Conference on Bioinformatics and Biomedicine*, pages 340–345, 2020.
- [44] P. van den Driessche. Spatial structure: patch models. In *Mathematical Epidemiology*, chapter 7, pages 179–189. Springer-Verlag, Berlin, 2008.
- [45] Z. Xu, F. Yan, and Y. Qi. Infinite Tucker decomposition: nonparametric Bayesian models for multiway data analysis. In *Proceedings of the 29th International Conference on International Conference on Machine Learning*, pages 1675–1682, Madison, Wisconsin, 2012. Omnipress.
- [46] Q. Zhao, G. Zhou, L. Zhang, and A. Cichocki. Tensor-variate Gaussian processes regression and its application to video surveillance. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1265–1269, 2014.

Appendix A. Summaries of predictive performance.

The following quantities are used to summarize emulator predictive performance. In each case, $\mu_f(\cdot)$ is the emulator mean, $\nu_f(\cdot)$ the emulator variance and $\{\tilde{\mathbf{x}}^{(k')}, f(\tilde{\mathbf{x}}^{(k')})\}_{k'=1}^{n'}$ is a set of test points (inputs and simulator outputs).

1. Mean Absolute Standardised Prediction Error (MASPE) [26]:

$$(A.1) \quad \frac{1}{n'} \sum_{k'=1}^{n'} \frac{|f(\tilde{\mathbf{x}}^{(k')}) - \mu_f(\tilde{\mathbf{x}}^{(k')})|}{\sqrt{\nu_f(\tilde{\mathbf{x}}^{(k')})}}.$$

2. Root Mean Squared Prediction Error (RMSPE) [2]:

$$(A.2) \quad \sqrt{\frac{1}{n'} \sum_{k'=1}^{n'} \{f(\tilde{\mathbf{x}}^{(k')}) - \mu_f(\tilde{\mathbf{x}}^{(k')})\}^2}.$$

3. Mean Generalised Entropy Score (MGES) [15]:

$$(A.3) \quad -\frac{1}{n'} \sum_{k'=1}^{n'} \left\{ \frac{[f(\tilde{\mathbf{x}}^{(k')}) - \mu_f(\tilde{\mathbf{x}}^{(k')})]^2}{\nu_f(\tilde{\mathbf{x}}^{(k')})} + \log(\nu_f(\tilde{\mathbf{x}}^{(k')})) \right\}.$$

MASPE is a measure of emulator validity; heuristically we expect this value to be around 1 (assuming normal errors this value should be $\sqrt{2/\pi}$ based on properties of the half-normal distribution). RMSPE permits comparison of emulator accuracy. MGES is larger (including corresponding sign) for approximations that are both accurate and valid.

Appendix B. Supporting comparison of the emulators.

B.1. Vectorized mean tensors. Vectorization of the mean tensor \mathcal{M} , applying definition (2.5) with the OPE mean (2.12), leads to a mean vector

$$(A.1) \quad \text{vec}(\mathcal{M}) = (\otimes_{i=0}^m G_i) \boldsymbol{\beta},$$

with

$$G_i = \begin{bmatrix} \mathbf{g}_i^\top(\mathbf{x}^{(1)}) \\ \vdots \\ \mathbf{g}_i^\top(\mathbf{x}^{(n)}) \end{bmatrix}.$$

Similarly, assuming the PPE mean (2.14) leads to a mean vector

$$(A.2) \quad \text{vec}(\mathcal{M}) = (G_0 \otimes I_r) \text{vec}(\mathcal{B}).$$

Hence, when vectorized via (2.9), both emulators can be cast in the outer product emulator framework; the mean vector takes the form of a linear predictor with both the model matrix and the variance-covariance matrix having Kronecker form and efficient computation can be achieved [39]. In fact, the independence of the parallel emulators in the PPE leads to even more efficient computation.

B.2. The role of spatio-temporal coregionalization. From Equation (2.10) it is clear that, conditional on the mean tensor $\mathcal{M}(\tilde{\mathbf{x}})$, the posterior predictive mean at new input $\tilde{\mathbf{x}}$ is independent of the coregionalization matrix $\Sigma = \otimes_{i=1}^m \Sigma_i$. In fact, for the PPE with a non-informative prior on regression parameters, it can be shown that this independence also holds unconditionally [18]. We restate the result here using the notation from this paper.

Lemma B.1. *Assume a TvGP implying (2.9) with vectorized mean tensor (A.2) and constant prior on $\text{vec}(\mathcal{B})$. The generalized least squares estimator for $\text{vec}(\mathcal{B})$, or equivalently the posterior mean, is independent of the coregionalization matrix $\Sigma = \otimes_{i=1}^m \Sigma_i$ and has the form*

$$\widehat{\text{vec}(\mathcal{B})} = \left[\left(G_0^\top K^{-1} G_0 \right)^{-1} G_0^\top K^{-1} \otimes I_r \right] \text{vec}(\mathcal{F}).$$

Proof. For the most general tensor PPE we have

$$\begin{aligned} \widehat{\text{vec}(\mathcal{B})} &= \left[\left(G_0^\top \otimes I_r \right) \left(K^{-1} \otimes \Sigma^{-1} \right) \left(G_0 \otimes I_r \right) \right]^{-1} \left(G_0^\top \otimes I_r \right) \left(K^{-1} \otimes \Sigma^{-1} \right) \text{vec}(\mathcal{F}) \\ &= \left[\left(G_0^\top K^{-1} G_0 \right)^{-1} \otimes \Sigma \right] \left(G_0^\top K^{-1} \otimes \Sigma^{-1} \right) \text{vec}(\mathcal{F}) \\ &= \left[\left(G_0^\top K^{-1} G_0 \right)^{-1} G_0^\top K^{-1} \otimes I_r \right] \text{vec}(\mathcal{F}), \end{aligned}$$

which does not involve Σ . ■

The form of the GLS estimator from Lemma B.1 also demonstrates the independence of the estimators for each output location.

In contrast, the GLS estimator for the tensor OPE does not simplify to such an extent, even with non-informative priors on the regression coefficients, as demonstrated via Lemma B.2.

Lemma B.2. *Assume a TvGP implying (2.9) with vectorized mean tensor (A.1) and constant prior on β . The generalized least squares estimator for β , or equivalently the posterior mean, depends on the coregionalization matrix $\Sigma = \otimes_{i=1}^m \Sigma_i$ and has the form*

$$\hat{\beta} = \left[\left(G_0^\top K^{-1} G_0 \right)^{-1} G_0^\top K^{-1} \otimes \left(G_{1:m}^\top \Sigma^{-1} G_{1:m} \right)^{-1} G_{1:m}^\top \Sigma^{-1} \right] \text{vec}(\mathcal{F}),$$

with $G_{1:m} = \otimes_{i=1}^m G_i$.

Proof. For the most general tensor OPE we have

$$\begin{aligned}
 \hat{\boldsymbol{\beta}} &= \left[\left(G_0^\top \otimes G_{1:m}^\top \right) (K^{-1} \otimes \Sigma^{-1}) (G_0 \otimes G_{1:m}) \right]^{-1} \left(G_0^\top \otimes G_{1:m}^\top \right) (K^{-1} \otimes \Sigma^{-1}) \text{vec}(\mathcal{F}) \\
 &= \left[\left(G_0^\top K^{-1} G_0 \right)^{-1} \otimes \left(G_{1:m}^\top \Sigma^{-1} G_{1:m} \right)^{-1} \right] \left(G_0^\top K^{-1} \otimes G_{1:m}^\top \Sigma^{-1} \right) \text{vec}(\mathcal{F}) \\
 &= \left[\left(G_0^\top K^{-1} G_0 \right)^{-1} G_0^\top K^{-1} \otimes \left(G_{1:m}^\top \Sigma^{-1} G_{1:m} \right)^{-1} G_{1:m}^\top \Sigma^{-1} \right] \text{vec}(\mathcal{F}).
 \end{aligned}$$

529 Hence the GLS equations are a Kronecker product of equations across the input and output regres-
 530 sors involving Σ . ■

531 Therefore, for a tensor OPE the mean prediction has the property of smoothing across output
 532 locations, which can be beneficial when there are strong output dependencies.

Appendix C. Additional figures. Here, we present some additional supporting figures for the two case studies.

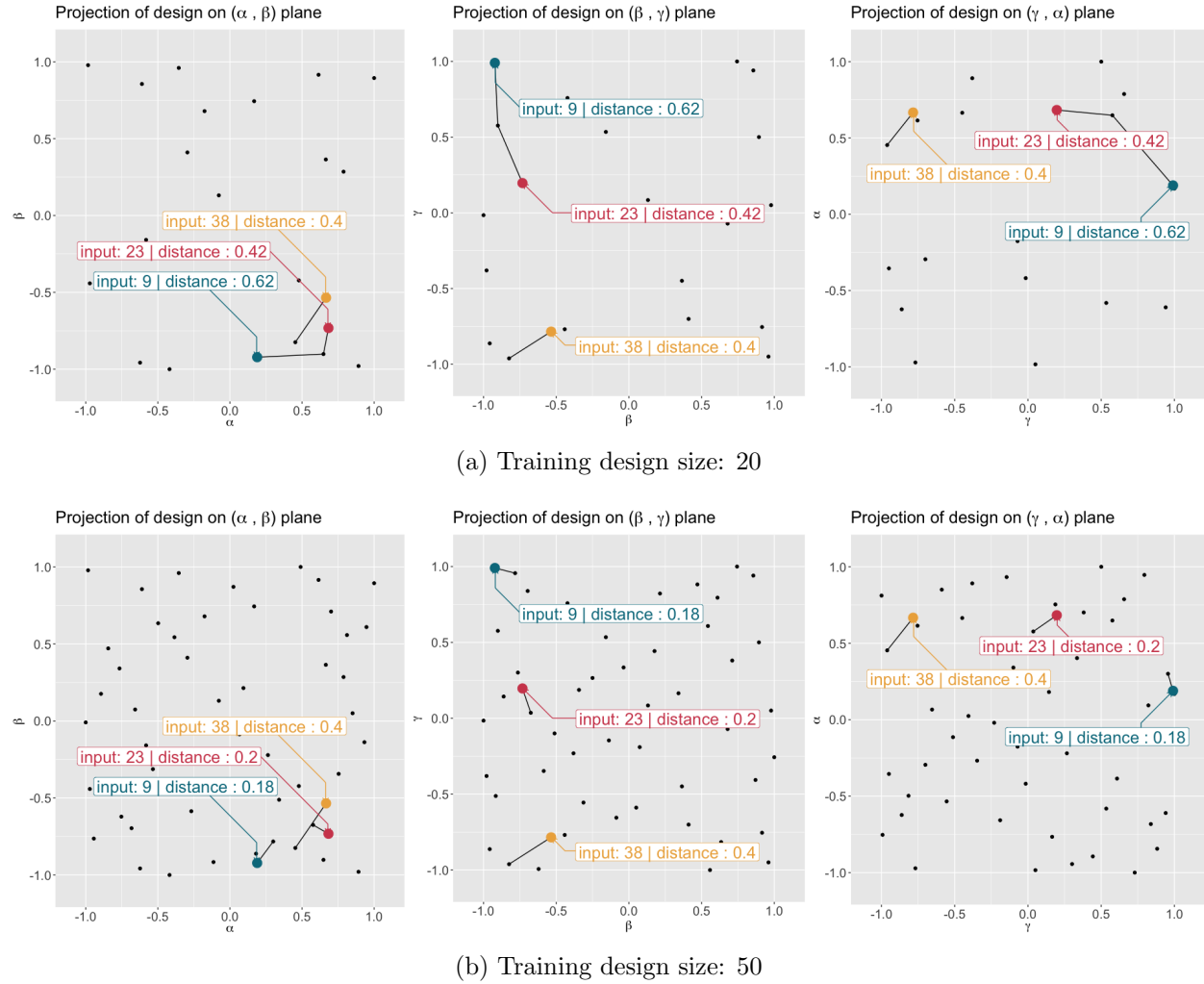


Figure A.1: Two-dimensional projections of the inputs for each of the test points presented in Figures 1 and 2 for the influenza simulator. Training design points are plotted as black dots. Each test point is connected to the nearest design point and the distance is stated in the label.

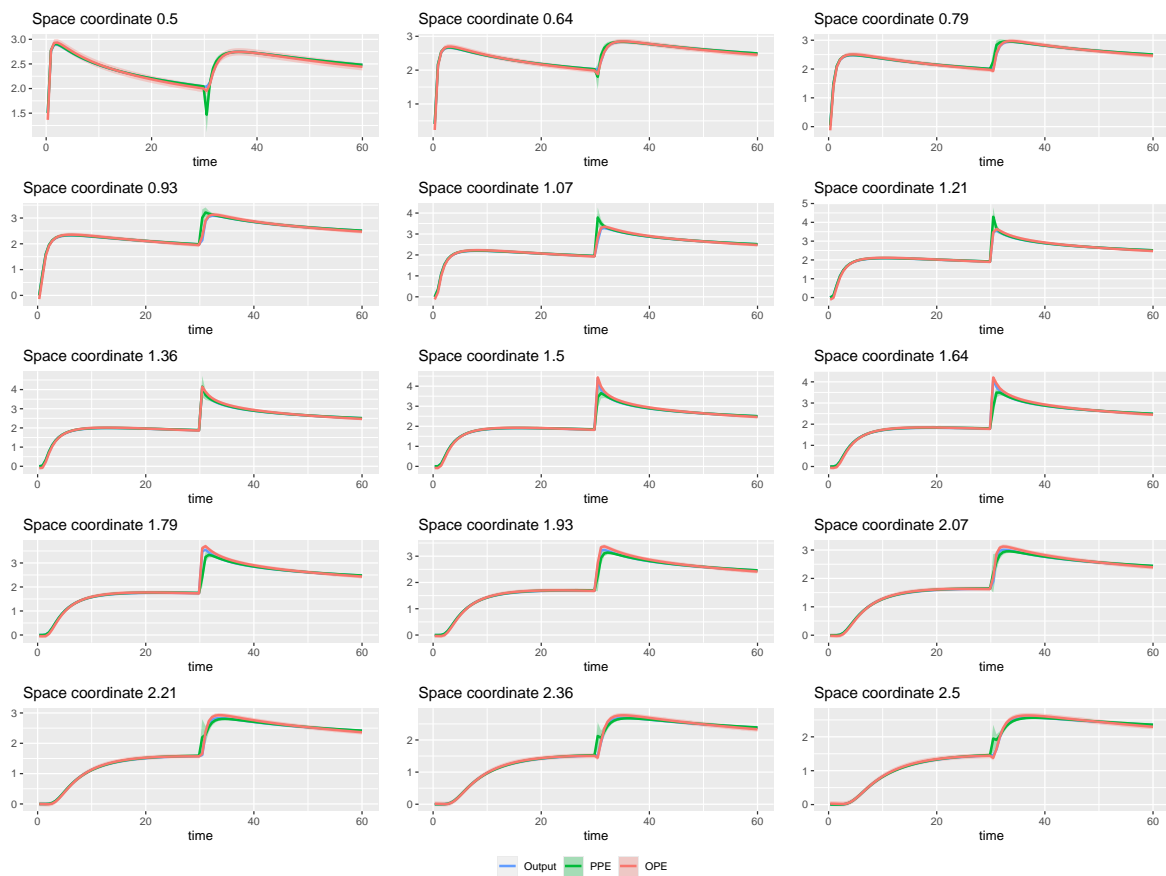


Figure A.2: Trace plots for the environmental simulator output, OPE and PPE mean predictions and uncertainty (± 3 standard deviations using $n = 50$ training points) for 15 spatial locations at input $x_1 = 10, x_2 = 1.505, x_3 = 30.1525, x_4 = 0.07$. Predictions and responses are on the log scale.