Gaussian processes for Bayesian inverse problems associated with linear partial differential equations

Tianming Bai[1,2], Aretha L. Teckentrup[1,2], Konstantinos C. Zygalakis[1,2]

August 31, 2023

Abstract

This work is concerned with the use of Gaussian surrogate models for Bayesian inverse problems associated with linear partial differential equations. A particular focus is on the regime where only a small amount of training data is available. In this regime the type of Gaussian prior used is of critical importance with respect to how well the surrogate model will perform in terms of Bayesian inversion. We extend the framework of Raissi et. al. (2017) to construct PDE-informed Gaussian priors that we then use to construct different approximate posteriors. A number of different numerical experiments illustrate the superiority of the PDE-informed Gaussian priors over more traditional priors.

1 Introduction

Combining complex mathematical models with observational data is an extremely challenging yet ubiquitous problem in the field of modern applied mathematics and data science. Inverse problems, where one is interested in learning inputs to a mathematical model such as physical parameters and initial conditions given partial and noisy observation of model outputs, are hence of frequent interest. Adopting a Bayesian approach [15, 32], we incorporate our prior knowledge on the inputs into a probability distribution, *the prior distribution*, and obtain a more accurate representation of the model inputs in the *posterior distribution*, which results from conditioning the prior distribution on the observed data.

The posterior distribution contains all the necessary information about the characteristics of our inputs. However, in most cases the posterior is unfortunately intractable and one needs to resort to sampling methods such as Markov

[1]School of Mathematics, University of Edinburgh, Edinburgh, Scotland, UK
[2]Maxwell Institute for Mathematical Sciences, Bayes Centre, 47 Potterrow, Edinburgh, Scotland, UK.

chain Monte Carlo (MCMC) [26, 3] to explore it. A major challenge in the application of MCMC methods to problems of practical interest is the large computational cost associated with numerically solving the mathematical model for a given set of the input parameters. Since the generation of each sample by an MCMC method requires a solve of the governing equations, and often millions of samples are required in practical applications, this process can quickly become very costly.

One way to deal with the challenge of full Bayesian inference for complex models is the use of surrogate models, also known as emulators, meta-models or reduced order models. In particular, instead of using the complex (and computationally expensive) model, one uses a simpler and computationally more efficient model to approximate the solution of the governing equations, which in turn is used to approximate the data likelihood. Within the statistics literature, the most commonly used type of surrogate model is a Gaussian process emulator [24, 30, 28, 16, 22, 14], but other types of surrogate models can also be used including projection-based methods [5], generalised Polynomial Chaos [35, 19], sparse grid collocation [1, 18] and adaptive subspace methods [9, 10].

In this paper, we focus on the use of Gaussian process surrogate models for approximating the posterior distribution in inverse problems, where the forward model relates to the solution of a linear partial differential equation (PDE). In particular, we consider two different ways of using the surrogate model, by emulating either the parameter-to-observation map or the negative log-likelihood. Convergence properties of the corresponding posterior approximations, as the number of design points N used to construct the surrogate model goes to infinity, have recently been studied in [31, 34, 13]. These results put the methodology on a firm theoretical footing, and show that the error in the approximate posterior distribution can be bounded by the corresponding error in the surrogate model. Furthermore, the error in the approximate posteriors tends to zero as N tends to infinity. However, when the forward model of interest is given by a complex model such as a PDE, one normally operates in a regime where only a very limited number of design points N can be used due to constraints on computational cost. This setting is less understood and is the setting of main interest in this paper.

With a small number of design points, different modelling choices made in the derivation of the approximate posterior can have a large effect on its accuracy. In particular, the choice of Gaussian prior distribution in the emulator is crucial, as it heavily influences its accuracy. Intuitively, we want to make the prior distribution as informative as possible, by incorporating known information about the underlying forward model. For example, an informed prior specially tailored to solving the forward problem in linear PDEs can be found in [23]. For incorporating more general constraints, we refer the reader to the recent review [33]. Other modelling choices that require careful consideration are whether we build a surrogate model for the parameter-to-observation map or the log-likelihood directly, and whether we use the full distribution of the emulator or only the mean (see e.g [31, 17]).

The focus of this paper is on computational aspects of the use of Gaussian

process surrogate models in PDE inverse problems, with particular emphasis on the setting where the number of design points is limited by computational constraints. The main contributions of this paper are the following:

1. We extend the PDE-informed Gaussian process priors from [23] to enable their use in inverse problems, which requires a Gaussian process prior as a function of both the spatial variable of the PDE and the unknown parameter(s).

2. By showing that the required gradients can be computed explicitly, we establish that gradient-based MCMC samplers such as the Metropolis-adjusted Langevin algorithm (MALA) can be used to efficiently sample from the approximate posterior distributions.

3. Using a range of numerical examples, we demonstrate the isolated effects of various modelling choices made, and thus offer valuable insights and guidance for practitioners. This includes choices on posterior approximation in the inverse problem (e.g. emulating the parameter-to-observation map or the log-likelihood) and on prior distributions for the Gaussian process emulator (e.g. black-box or PDE-constrained).

The rest of the paper is organised as follows. In Section 2 we set up notation with respect to the inverse problems of interest, as well as discuss the different kinds of posterior approximations that result from using Gaussian surrogate models for the data-likelihood. We then proceed in Section 3 to present our main methodology, discussing how can one blend better-informed Gaussian surrogate models with inverse problems as well as presenting the MCMC algorithm that we use. A number of different numerical experiments that illustrate the computational benefits of our approach are then presented in Section 4, while finally Section 5 provides a summary and discussion of the main results.

2 Preliminaries

We now give more details about the type of inverse problems that we consider in this paper as well as discuss different aspects of Gaussian emulators and the corresponding type of approximate posteriors that we consider in this work. At the end of this section, we summarise in Table 1 all the different notations introduced in this section.

2.1 PDE Inverse problems

Consider the linear PDE

$$\mathcal{L}^{\boldsymbol{\theta}} u(\mathbf{x}) = f(\mathbf{x}), \qquad \mathbf{x} \in D, \tag{1a}$$

$$\mathcal{B} u(\mathbf{x}) = g(\mathbf{x}), \qquad \mathbf{x} \in \partial D, \tag{1b}$$

posed on a computational domain $D \subseteq \mathbb{R}^{d_{\mathbf{x}}}$, where $\mathcal{L}^{\boldsymbol{\theta}}$ denotes a linear differential operator depending on parameters $\boldsymbol{\theta} \in \mathcal{T} \subseteq \mathbb{R}^{d_{\boldsymbol{\theta}}}$ and the linear operator $\mathcal{B}$

incorporates boundary conditions. The inverse problem of interest in this paper is to infer the parameters $\boldsymbol{\theta}$ from the noisy data $\mathbf{y} \in \mathbb{R}^{d_\mathbf{y}}$ given by

$$\mathbf{y} = \mathcal{G}_X(\boldsymbol{\theta}) + \boldsymbol{\eta}, \tag{2}$$

where $X = \{\mathbf{x}_1, \cdots, \mathbf{x}_{d_\mathbf{y}}\} \subset \overline{D}$ are the spatial points where we observe the solution u of our PDE, $\mathcal{G}_X : \mathcal{T} \to \mathbb{R}^{d_\mathbf{y}}$ is the *parameter-to-observation map* defined by $\mathcal{G}_X(\boldsymbol{\theta}) = \{u(\mathbf{x}_j; \boldsymbol{\theta})\}_{j=1}^{d_\mathbf{y}}$, and $\boldsymbol{\eta} \sim \mathcal{N}(0, \Gamma_\eta)$ is an additive Gaussian noise term with covariance matrix $\Gamma_\eta = \sigma_\eta^2 I_{d_\mathbf{y}}$. Note that the assumption of Gaussianity and diagonal noise covariance is done for simplicity, but these assumptions can be relaxed [17]. Likewise, the methodology generalises straightforwardly to general bounded linear observation operators applied to the PDE solution u.

To solve the inverse problem we will adopt a Bayesian approach [32]. That is, prior to observing the data $\mathbf{y}$, $\boldsymbol{\theta}$ is assumed to be distributed according to a prior density $\pi_0(\boldsymbol{\theta})$, and we are interested in the updated posterior density $\pi(\boldsymbol{\theta}|\mathbf{y})$. From (2) we have $\mathbf{y}|\boldsymbol{\theta} \sim \mathcal{N}(\mathcal{G}_X(\boldsymbol{\theta}), \Gamma_\eta)$, so the *likelihood* is

$$L(\mathbf{y}|\boldsymbol{\theta}) \propto \exp\left(-\frac{1}{2}\|\mathcal{G}_X(\boldsymbol{\theta}) - \mathbf{y}\|_{\Gamma_\eta}^2\right) \coloneqq \exp\left(-\Phi(\boldsymbol{\theta}, \mathbf{y})\right), \tag{3}$$

where the function $\Phi : \mathcal{T} \times \mathbb{R}^{d_\mathbf{y}} \to \mathbb{R}$ is called the *negative log-likelihood* or *potential* and $\|\mathbf{z}\|_{\Gamma_\eta} = \mathbf{z}^\mathrm{T}\Gamma_\eta^{-1}\mathbf{z}$ denotes the norm weighted by Γ_η^{-1}. Then by Bayes' formula we have

$$\pi(\boldsymbol{\theta}|\mathbf{y}) \propto L(\mathbf{y}|\boldsymbol{\theta})\pi_0(\boldsymbol{\theta}). \tag{4}$$

The posterior distribution $\pi(\boldsymbol{\theta}|\mathbf{y})$ is in general intractable, and we need to resort to sampling methods such as MCMC to extract information from it. However, generating a sample typically involves evaluating the likelihood and hence the solution of the PDE (1), which can be prohibitively costly. This motivates the use of surrogate models to emulate the PDE solution, which in turn is used to approximate the posterior and hence accelerate the sampling process.

2.2 Gaussian processes

Gaussian process regression (GPR) is a flexible non-parametric model for Bayesian inference [24]. In particular our starting point for approximating an arbitrary function $\mathbf{g} : \mathcal{T} \to \mathbb{R}^d$, for some $d \in \mathbb{N}$, in the absence of any observations is the following Gaussian process prior

$$\mathbf{g}_0(\boldsymbol{\theta}) \sim \mathrm{GP}(\mathbf{m}(\boldsymbol{\theta}), K(\boldsymbol{\theta}, \boldsymbol{\theta}')), \tag{5}$$

where $\mathbf{m} : \mathcal{T} \to \mathbb{R}^d$ is a mean function and $K(\boldsymbol{\theta}, \boldsymbol{\theta}') : \mathcal{T} \times \mathcal{T} \to \mathbb{R}^{d \times d}$ is the matrix-valued covariance function which represents the covariance between the different entries of $\mathbf{g}$ evaluated at $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$. When emulating the forward map the function $\mathbf{g}$ corresponds to the PDE solution evaluated at $d_\mathbf{y}$ different spatial points, and hence $d = d_\mathbf{y}$. In contrast when emulating directly the log-likelihood $d = 1$. Furthermore, the matrix $K(\boldsymbol{\theta}, \boldsymbol{\theta}')$ is often assumed to take the form

$$K(\boldsymbol{\theta}, \boldsymbol{\theta}') = k(\boldsymbol{\theta}, \boldsymbol{\theta}')I_d \tag{6}$$

for some scalar-valued covariance function $k(\boldsymbol{\theta}, \boldsymbol{\theta}') : \mathcal{T} \times \mathcal{T} \to \mathbb{R}$, implying that the entries of $\mathbf{g}$ are independent. As we will see later better emulators can be constructed by relaxing this independence assumption.

The mean function and the covariance function fully characterise our Gaussian prior. A typical choice for $\mathbf{m}$ is to set it to zero, while common choices for the covariance function $k(\boldsymbol{\theta}, \boldsymbol{\theta}')$ include the squared exponential covariance function

$$k(\boldsymbol{\theta}, \boldsymbol{\theta}') = \sigma^2 \exp\left(-\frac{\left\|\boldsymbol{\theta} - \boldsymbol{\theta}'\right\|^2}{2l^2}\right),$$

and the Matérn covariance functions

$$k(\boldsymbol{\theta}, \boldsymbol{\theta}') = \frac{\sigma^2}{\Gamma(\nu)2^{\nu-1}}\left(\sqrt{2\nu}\frac{\left\|\boldsymbol{\theta} - \boldsymbol{\theta}'\right\|}{l}\right)^{\nu} B_\nu\left(\sqrt{2\nu}\frac{\left\|\boldsymbol{\theta} - \boldsymbol{\theta}'\right\|}{l}\right).$$

For both kernels, the hyperparameter $\sigma^2 > 0$ governs the magnitude of the covariance and the hyperparameter $l > 0$ governs the length-scale at which the entries of $\mathbf{g}_0(\boldsymbol{\theta})$ and $\mathbf{g}_0(\boldsymbol{\theta}')$ are correlated. For the Matérn covariance function the smoothness of the entries of $\mathbf{g}_0$ depends on the positive hyper-parameter ν, while in the limit $\nu \to \infty$ we obtain the squared exponential covariance function which gives rise to infinitely differentiable sample paths for $\mathbf{g}_0$.

Now suppose that we are given data in the form of N distinct design points $\Theta = \{\boldsymbol{\theta}^i\}_{i=1}^N \in \mathbb{R}^{d_{\boldsymbol{\theta}} \times N}$ with corresponding function values

$$\mathbf{g}(\Theta) := [\mathbf{g}(\boldsymbol{\theta}^1); \cdots ; \mathbf{g}(\boldsymbol{\theta}^N)] \in \mathbb{R}^{(d_{\mathbf{y}} \times N) \times 1}.$$

Since we have assumed that the multi-output function $\mathbf{g}_0$ is a Gaussian process, the matrix vector

$$[\mathbf{g}_0(\boldsymbol{\theta}^1); \cdots ; \mathbf{g}_0(\boldsymbol{\theta}^N); \mathbf{g}_0(\tilde{\boldsymbol{\theta}})] \in \mathbb{R}^{(d_{\mathbf{y}} \times (N+1)) \times 1}$$

for any test point $\tilde{\boldsymbol{\theta}}$ follows a multivariate Gaussian distribution. The conditional distribution of $\mathbf{g}_0(\tilde{\boldsymbol{\theta}})$ given the set of values $\mathbf{g}(\Theta)$ is then again Gaussian with mean and covariance given by the standard formulas for the conditioning of Gaussian random variables [24]. In particular, if we denote with $\mathbf{g}^N$ the Gaussian process (5) conditioned on the values $\mathbf{g}(\Theta)$ we have

$$\mathbf{g}^N(\boldsymbol{\theta}) \sim \mathrm{GP}(\mathbf{m}_N^{\mathbf{g}}(\boldsymbol{\theta}), K_N(\boldsymbol{\theta}, \boldsymbol{\theta}')) \tag{7}$$

where the predictive mean vector $\mathbf{m}_N^{\mathbf{g}}$ and the predictive covariance matrix $K_N(\boldsymbol{\theta}, \boldsymbol{\theta}')$ are given by

$$\mathbf{m}_N^{\mathbf{g}}(\boldsymbol{\theta}) = \mathbf{m}(\boldsymbol{\theta}) + K(\boldsymbol{\theta}, \Theta)K(\Theta, \Theta)^{-1}\left(\mathbf{g}(\Theta) - \mathbf{m}(\Theta)\right) \tag{8}$$

$$K_N(\boldsymbol{\theta}, \boldsymbol{\theta}') = K(\boldsymbol{\theta}, \boldsymbol{\theta}') - K(\boldsymbol{\theta}, \Theta)K(\Theta, \Theta)^{-1}K(\boldsymbol{\theta}', \Theta)^T, \tag{9}$$

with $\mathbf{m}(\Theta) = [\mathbf{m}(\boldsymbol{\theta}^1); \cdots ; \mathbf{m}(\boldsymbol{\theta}^N)] \in \mathbb{R}^{(d_{\mathbf{y}} \times N) \times 1}$, $K(\boldsymbol{\theta}, \Theta) = [K(\boldsymbol{\theta}, \boldsymbol{\theta}^1), \ldots, K(\boldsymbol{\theta}, \boldsymbol{\theta}^N)] \in \mathbb{R}^{d_{\mathbf{y}} \times (d_{\mathbf{y}} \times N)}$ and

$$K(\Theta,\Theta) = \begin{bmatrix} K(\boldsymbol{\theta}^1,\boldsymbol{\theta}^1) & \dots & K(\boldsymbol{\theta}^1,\boldsymbol{\theta}^N) \\ \vdots & & \vdots \\ K(\boldsymbol{\theta}^N,\boldsymbol{\theta}^1) & \dots & K(\boldsymbol{\theta}^N,\boldsymbol{\theta}^N) \end{bmatrix} \in \mathbb{R}^{(d_{\mathbf{y}}\times N)\times(d_{\mathbf{y}}\times N)}$$

To avoid ambiguity in the notation, we use regular font for scalar values, bold font for vector values, and capital font for matrices (details in Table 1).

2.3 Gaussian emulators and approximate posteriors

We now discuss two different approaches for constructing a Gaussian emulator and using it for approximating the posterior of interest. The first approach constructs an emulator for the forward map $\mathcal{G}_X$, while the second approach is based on constructing an emulator directly for the log-likelihood.

2.3.1 Emulating the forward map

Given the data set $\mathcal{G}_X(\Theta)$, we can now proceed with building our Gaussian process emulation for the forward map $\mathcal{G}_X$. One then needs to decide how to incorporate the emulation for the construction of an approximate posterior. In particular, depending on what type of information we plan to utilize, different approximations will be obtained. If we use its predictive mean $\mathbf{m}_N^{\mathcal{G}_X}$ as a point estimator of the forward map $\mathcal{G}_X$, we obtain

$$\pi_{\text{mean}}^{N,\mathcal{G}_X}(\boldsymbol{\theta}|\mathbf{y}) \propto \exp\left(-\frac{1}{2}\|\mathbf{m}_N^{\mathcal{G}_X}(\boldsymbol{\theta}) - \mathbf{y}\|^2_{\Gamma_\eta}\right)\pi_0(\boldsymbol{\theta}). \tag{10}$$

Alternatively, we can try to exploit the full information given by the Gaussian process by incorporating its variance in the posterior approximation. A natural way to do this is to consider the following approximation[1]:

$$\begin{aligned}\pi_{\text{marginal}}^{N,\mathcal{G}_X}(\boldsymbol{\theta}|\mathbf{y}) &\propto \mathbb{E}\left(\exp\left(-\frac{1}{2}\|\mathcal{G}_X^N(\boldsymbol{\theta}) - \mathbf{y}\|^2_{\Gamma_\eta}\right)\pi_0(\boldsymbol{\theta})\right) \\ &\propto \left(\frac{\exp\left(-\frac{1}{2}\|\mathbf{m}_N^{\mathcal{G}_X}(\boldsymbol{\theta}) - \mathbf{y}\|^2_{(K_N(\boldsymbol{\theta},\boldsymbol{\theta})+\Gamma_\eta)}\right)}{\sqrt{(2\pi)^{d_{\mathbf{y}}}\det\left(K_N(\boldsymbol{\theta},\boldsymbol{\theta})+\Gamma_\eta\right)}}\right)\pi_0(\boldsymbol{\theta}),\end{aligned} \tag{11}$$

Comparing (11) with (10), the likelihood function in the marginal approximation is Gaussian with additional uncertainty $K_N(\boldsymbol{\theta},\boldsymbol{\theta})$ from the emulator included into its covariance matrix. Hence, for a fixed parameter $\boldsymbol{\theta}$, the likelihood function in (11) will be less concentrated due to the variance inflation. When the magnitude of $K_N(\boldsymbol{\theta},\boldsymbol{\theta})$ is small compared to that of Γ_η, the marginal approximation will be similar to the mean-based approximation.

[1]The derivation of (11) results from the fact that the convolution of two Gaussian measures is Gaussian. A detailed derivation can be found in Appendix for completeness, the formula was also derived in [7, 6].

2.3.2 Emulating the log-likelihood

Another way of building the emulator is to model the potential function Φ directly. We can convert the data set $\mathcal{G}_X(\Theta)$ into a data set of negative log-likelihood $\mathbf{\Phi}(\Theta) = \{\Phi(\boldsymbol{\theta}^i, \mathbf{y})\}_{i=1}^N$. Again, if we only include the mean of the Gaussian process emulator the posterior approximation becomes

$$\pi_{\text{mean}}^{N,\Phi}(\boldsymbol{\theta}|\mathbf{y}) \propto \exp\left(-m_N^{\Phi}(\boldsymbol{\theta})\right) \pi_0(\boldsymbol{\theta}), \tag{12}$$

while, in a similar fashion to the forward map emulation, we can take into account the covariance of our emulator to obtain the approximate posterior

$$\begin{aligned}\pi_{\text{marginal}}^{N,\Phi}(\boldsymbol{\theta}|\mathbf{y}) &\propto \mathbb{E}\left((\exp\left(-\Phi^N(\boldsymbol{\theta})\right))\pi_0(\boldsymbol{\theta})\right) \\ &\propto \exp\left(-m_N^{\Phi}(\boldsymbol{\theta}) + \frac{1}{2}k_N(\boldsymbol{\theta},\boldsymbol{\theta})\right)\pi_0(\boldsymbol{\theta}). \end{aligned} \tag{13}$$

The derivation of (13) is similar to that of (11). Note that in this case, the following relationship holds between the two approximate posteriors

$$\pi_{\text{marginal}}^{N,\Phi}(\boldsymbol{\theta}|\mathbf{y}) \propto \pi_{\text{mean}}^{N,\Phi}(\boldsymbol{\theta}|\mathbf{y}) \exp\left(\frac{1}{2}k_N(\boldsymbol{\theta},\boldsymbol{\theta})\right),$$

which again illustrates a form of variance inflation for the marginal posterior approximation.

In summary, we have two methods for approximating posteriors: the mean-based approximation and the marginal approximation; and we have two types of emulators: the forward map emulator and the potential function emulator; thus by combination we have four types of approximations in total. The convergence properties of all these approximate posteriors where the subject of study in [31, 34, 13], where it was proved under suitable assumptions that all of them converge to the true posterior as $N \to \infty$. However, in the case of small N, the difference between the approximate posteriors could be large and which one we choose is important. Furthermore, the type of Gaussian process emulator used plays an even bigger role in this case, and one would like to use a Gaussian prior that is as informative as possible. We discuss how to do this in the next section.

3 Methodology

Having described the different types of posterior approximations we will consider, in this section we discuss different modelling approaches for the prior distribution used in our Gaussian emulators. In doing this it is important to note that the function that we are interested to emulate, in this case the forward map $\mathcal{G}_X(\boldsymbol{\theta})$, depends not only on the parameters $\boldsymbol{\theta}$ of our PDE, but also on the location of the spatial observations. Thus in terms of modelling, one would like to take this into account and build spatial correlation explicitly into the prior covariance. This can be done in two different ways, the first by prescribing some explicit form of spatial correlation, and the second by using the fact that we

Symbol	Description
$\boldsymbol{\theta}$	Unknown parameter in PDE
$\mathcal{T}$	Space of unknown parameter
$\mathbf{y}$	Discrete observation of PDE solution
$d_{\boldsymbol{\theta}}$, $d_{\mathbf{y}}$, $d_{\mathbf{x}}$	Dimension of vector space
$\boldsymbol{\eta}$, Γ_η, σ_η^2	Gaussian noise $\boldsymbol{\eta}$ with zero mean and covariance matrix $\Gamma_\eta = \sigma_\eta^2 I_{d_{\mathbf{y}}}$
$X = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{d_{\mathbf{y}}}\}$	Set of spatial points corresponding to the observation $\boldsymbol{y}$
$\mathcal{G}_X$	$\mathcal{G}_X : \mathcal{T} \to \mathbb{R}^{d_{\mathbf{y}}}$ parameter-to-observation map
$\pi(\boldsymbol{\theta}\|\mathbf{y})$	Posterior
$L(\mathbf{y}\|\boldsymbol{\theta})$	Likelihood
$\pi_0(\boldsymbol{\theta})$	Prior
$\Phi(\boldsymbol{\theta}, \mathbf{y})$	Negative log-likelihood (or potential)
σ^2, l	Hyper-parameters in Gaussian covariance function (variance σ^2 and length-scale l)
$\mathrm{GP}(\mathbf{m}(\boldsymbol{\theta}), K(\boldsymbol{\theta}, \boldsymbol{\theta}'))$	Gaussian process with mean function $\mathbf{m}(\boldsymbol{\theta})$ and matrix-valued covariance function $K(\boldsymbol{\theta}, \boldsymbol{\theta}')$
$k(\boldsymbol{\theta}, \boldsymbol{\theta}')$	Scalar-valued covariance function
$\mathcal{G}_X(\Theta)$	Training data set of function values at $\Theta = \{\boldsymbol{\theta}^i\}_{i=1}^N \in \mathbb{R}^{d_{\boldsymbol{\theta}} \times N}$
$\mathcal{G}_X^N(\boldsymbol{\theta})$	Gaussian process conditioned on data $\mathcal{G}_X(\Theta)$
$\mathbf{m}_N^{\mathcal{G}_X}(\boldsymbol{\theta})$, $K_N(\boldsymbol{\theta}, \boldsymbol{\theta}')$	Predictive mean and predictive covariance of $\mathcal{G}_X^N(\boldsymbol{\theta})$
$\mathcal{L}_{\mathbf{x}}^{\boldsymbol{\theta}}$	Differential operator of PDE with parameter $\boldsymbol{\theta}$
u, f	PDE solution u and sourcing term f
$\pi_{\text{mean}}^{N,\mathcal{G}_X}$, $\pi_{\text{mean}}^{N,\mathcal{G}_X,s}$, $\pi_{\text{mean}}^{N,\mathcal{G}_X,\text{PDE}}$	Mean-based posterior with baseline, spatially correlated and PDE-constrained emulator
$\pi_{\text{marginal}}^{N,\mathcal{G}_X}$, $\pi_{\text{marginal}}^{N,\mathcal{G}_X,s}$, $\pi_{\text{marginal}}^{N,\mathcal{G}_X,\text{PDE}}$	Marginal posterior with baseline, spatially correlated and PDE-constrained emulator
$\Phi(\Theta)$	Training data set of potential function values at Θ
$\Phi^N(\boldsymbol{\theta})$	Gaussian process conditioned on data $\Phi(\Theta)$
$m_N^\Phi(\boldsymbol{\theta})$, $k_N(\boldsymbol{\theta}, \boldsymbol{\theta}')$	Predictive mean and covariance of Φ^N
$\pi_{\text{mean}}^{N,\Phi}$, $\pi_{\text{marginal}}^{N,\Phi}$	Mean-based and marginal posterior with emulation of potential function
$k_p(\boldsymbol{\theta}, \boldsymbol{\theta}')$, $k_s(\mathbf{x}, \mathbf{x}')$	Scalar-valued covariance function for parameter and spatial coordinate
$K_p(\boldsymbol{\theta}, \boldsymbol{\theta}')$, $K_s(\mathbf{x}, \mathbf{x}')$	Matrix-valued covariance function for parameter and spatial coordinate

Table 1: The list of symbols and notations used in this paper.

know that our forward map is associated with the solution of a linear PDE. We do this in Section 3.1. It is important to note that in both cases it is possible to calculate the gradients with respect to the parameters $\boldsymbol{\theta}$ in a closed form, which can then be used to sample from the approximate posterior distributions using gradient-based MCMC methods such as MALA. We discuss this in more detail in Section 3.2.

3.1 Correlated and PDE-informed priors

We now discuss two different approaches to incorporate spatial correlation in our prior covariance function for the forward map $\mathcal{G}_X(\boldsymbol{\theta})$. Even though this is a function from the parameter space $\mathcal{T}$ to the observation space $\mathbb{R}^{d_{\mathbf{y}}}$, for introducing more complicated spatial correlation it is useful to think first about the PDE solution $u(\boldsymbol{\theta}, \mathbf{x})$ as a function from $\mathcal{T} \times \overline{D}$ to $\mathbb{R}$. We introduce the prior covariance function $k((\boldsymbol{\theta}, \mathbf{x}), (\boldsymbol{\theta}', \mathbf{x}'))$ for $u(\boldsymbol{\theta}, \mathbf{x})$, and choose a separable model

$$k((\boldsymbol{\theta}, \mathbf{x}), (\boldsymbol{\theta}', \mathbf{x}')) = k_p(\boldsymbol{\theta}, \boldsymbol{\theta}')k_s(\mathbf{x}, \mathbf{x}'), \tag{14}$$

where k_p and k_s are the covariance functions for the parameters $\boldsymbol{\theta}$ and the spatial points $\mathbf{x}$ respectively. Separable kernels are a common modeling assumption in Gaussian processes. The resulting covariance function will have a high value only if the kernels for both variables have a high value.

Using the fact that the forward map $\mathcal{G}_X$ relates to the point-wise evaluation of the function $u(\boldsymbol{\theta}, \mathbf{x})$ for $\mathbf{x} \in X$, and assuming zero mean, we then have the Gaussian prior

$$\mathcal{G}_X(\boldsymbol{\theta}) \sim \mathrm{GP}(0, K(\boldsymbol{\theta}, \boldsymbol{\theta}')), \tag{15}$$

with

$$K(\boldsymbol{\theta}, \boldsymbol{\theta}') = k_p(\boldsymbol{\theta}, \boldsymbol{\theta}')K_s(X, X),$$

where K_s is the covariance matrix with entries $(K_s(X, X))_{i,j} = k_s(\mathbf{x}_i, \mathbf{x}_j)$, $\mathbf{x}_i, \mathbf{x}_j \in X$. This prior can then be updated to a posterior by conditioning on data $\mathcal{G}_X(\Theta)$ as in Section 2.2, which gives

$$\mathcal{G}_X(\boldsymbol{\theta})|\mathcal{G}_X(\Theta) \sim \mathrm{GP}(\mathbf{m}_N^{\mathcal{G}_X}(\boldsymbol{\theta}), K_N(\boldsymbol{\theta}, \boldsymbol{\theta}')), \tag{16}$$

with

$$\begin{aligned}
\mathbf{m}_N^{\mathcal{G}_X}(\boldsymbol{\theta}) &= K_{uu}(\boldsymbol{\theta}, \Theta)K_{uu}(\Theta, \Theta)^{-1}\mathcal{G}_X(\Theta), \\
K_N(\boldsymbol{\theta}, \boldsymbol{\theta}') &= K(\boldsymbol{\theta}, \boldsymbol{\theta}') - K_{uu}(\boldsymbol{\theta}, \Theta)K_{uu}(\Theta, \Theta)^{-1}K(\boldsymbol{\theta}', \Theta),
\end{aligned}$$

and

$$K_{uu}(\Theta, \Theta) = \{k_p(\boldsymbol{\theta}^i, \boldsymbol{\theta}^j)K_s(X, X)\} \in \mathbb{R}^{Nd_{\mathbf{y}} \times Nd_{\mathbf{y}}}, \quad \text{similarly} \quad K_{uu}(\boldsymbol{\theta}, \Theta) \in \mathbb{R}^{d_{\mathbf{y}} \times Nd_{\mathbf{y}}}.$$

The second way of introducing spatial correlation is to explicitly take into account that the forward map is related to a PDE solution. Given the PDE system

$$\begin{aligned}
\mathcal{L}^{\boldsymbol{\theta}}u(\mathbf{x}) &= f(\mathbf{x}), \qquad \mathbf{x} \in D, \\
\mathcal{B}u(\mathbf{x}) &= g(\mathbf{x}), \qquad \mathbf{x} \in \partial D,
\end{aligned}$$

as in Section 2, we can build a joint prior between u, f and g. In particular, if we take fixed points $\mathbf{x}, \mathbf{x}_f \in D$ and $\mathbf{x}_b \in \partial D$ we have that

$$\begin{bmatrix} u(\boldsymbol{\theta}, \mathbf{x}) \\ g(\boldsymbol{\theta}, \mathbf{x}_b) \\ f(\boldsymbol{\theta}, \mathbf{x}_f) \end{bmatrix} \sim \mathrm{GP}\left(\mathbf{0}, k_p(\boldsymbol{\theta}, \boldsymbol{\theta}') \begin{bmatrix} k_s(\mathbf{x}, \mathbf{x}) & \mathcal{B}k_s(\mathbf{x}, \mathbf{x}_b) & \mathcal{L}^{\boldsymbol{\theta}'} k_s(\mathbf{x}, \mathbf{x}_f) \\ \mathcal{B}k_s(\mathbf{x}_b, \mathbf{x}) & \mathcal{B}\mathcal{B}k_s(\mathbf{x}_b, \mathbf{x}_b) & \mathcal{B}\mathcal{L}^{\boldsymbol{\theta}'} k_s(\mathbf{x}_b, \mathbf{x}_f) \\ \mathcal{L}^{\boldsymbol{\theta}} k_s(\mathbf{x}_f, \mathbf{x}_b) & \mathcal{L}^{\boldsymbol{\theta}}\mathcal{B}k_s(\mathbf{x}_f, \mathbf{x}_b) & \mathcal{L}^{\boldsymbol{\theta}}\mathcal{L}^{\boldsymbol{\theta}'} k_s(\mathbf{x}_f, \mathbf{x}_f) \end{bmatrix} \right), \tag{17}$$

where the above is a Gaussian process as a function of $\boldsymbol{\theta}$, and we have used known properties of linear operators applied to Gaussian processes (see e.g. [20]) in the derivation. The idea of a joint prior between u and f was also used in [23, 29], with the crucial difference that u and f were considered as functions of the spatial variable $\mathbf{x}$ only. In the context of inverse problems and emulators as considered in this work, we instead explicitly model the dependency of u on $\boldsymbol{\theta}$, which requires an extension of the methodology.

Now as in the spatially correlated case, we can use the formula (17) to obtain a (joint) prior for $\mathcal{G}_X(\boldsymbol{\theta})$. More precisely, we have

$$\begin{bmatrix} \mathcal{G}_X(\boldsymbol{\theta}) \\ g(\boldsymbol{\theta}, X_g) \\ f(\boldsymbol{\theta}, X_f) \end{bmatrix} \sim \mathrm{GP}\left(\mathbf{0}, K(\boldsymbol{\theta}, \boldsymbol{\theta}')\right), \tag{18}$$

where

$$K(\boldsymbol{\theta}, \boldsymbol{\theta}') = k_p(\boldsymbol{\theta}, \boldsymbol{\theta}') \begin{bmatrix} K_s(X, X) & \mathcal{B}K_s(X, X_g) & \mathcal{L}^{\boldsymbol{\theta}'} K_s(X, X_f) \\ \mathcal{B}K_s(X_g, X) & \mathcal{B}\mathcal{B}K_s(X_g, X_g) & \mathcal{B}\mathcal{L}^{\boldsymbol{\theta}'} K_s(X_g, X_f) \\ \mathcal{L}^{\boldsymbol{\theta}} K_s(X_f, X) & \mathcal{L}^{\boldsymbol{\theta}}\mathcal{B}K_s(X_f, X_g) & \mathcal{L}^{\boldsymbol{\theta}}\mathcal{L}^{\boldsymbol{\theta}'} K_s(X_f, X_f) \end{bmatrix}$$

and $X_g \subset \partial D$ and $X_f \subset D$ are collections of d_g and d_f points at which g and f have been evaluated, respectively. Note that the marginal prior placed on $\mathcal{G}_X$ is the same as in (15).

We can then condition the joint Gaussian process prior (18) as in Section 2.2 on observations $\mathbf{g}(\Theta)$, where now

$$\mathbf{g} = \begin{bmatrix} \mathcal{G}_X(\cdot) \\ g(\cdot, X_g) \\ f(\cdot, X_f) \end{bmatrix} : \mathcal{T} \to \mathbb{R}^{d_{\mathbf{y}} + d_g + d_f}.$$

After a re-ordering of the observations $\mathbf{g}(\Theta)$, this results in the conditional distribution

$$\mathbf{g}(\boldsymbol{\theta}) | \mathbf{g}(\Theta) \sim \mathrm{GP}\left(\mathbf{m}_N^{\mathbf{g}}(\boldsymbol{\theta}), K_N(\boldsymbol{\theta}, \boldsymbol{\theta}')\right),$$

where

$$\begin{aligned} \mathbf{m}_N^{\mathbf{g}}(\boldsymbol{\theta}) &= \tilde{K}(\boldsymbol{\theta}, \Theta)\tilde{K}(\Theta, \Theta)^{-1}\mathbf{g}(\Theta), \\ K_N^{\mathbf{g}}(\boldsymbol{\theta}, \boldsymbol{\theta}') &= K(\boldsymbol{\theta}, \boldsymbol{\theta}') - \tilde{K}(\boldsymbol{\theta}, \Theta)\tilde{K}(\Theta, \Theta)^{-1}\tilde{K}(\boldsymbol{\theta}', \Theta)^{\mathrm{T}}, \end{aligned}$$

with $K(\boldsymbol{\theta}, \boldsymbol{\theta}') = k_p(\boldsymbol{\theta}, \boldsymbol{\theta}')K_s(X, X)$ as before and

$$\tilde{K}(\boldsymbol{\theta}, \Theta) = \begin{bmatrix} K_{uu}(\boldsymbol{\theta}, \Theta) & K_{ug}(\boldsymbol{\theta}, \Theta) & K_{uf}(\boldsymbol{\theta}, \Theta) \\ K_{ug}^{\mathrm{T}}(\boldsymbol{\theta}, \Theta) & K_{gg}(\boldsymbol{\theta}, \Theta) & K_{gf}(\boldsymbol{\theta}, \Theta) \\ K_{uf}^{\mathrm{T}}(\boldsymbol{\theta}, \Theta) & K_{gf}^{T}(\boldsymbol{\theta}, \Theta) & K_{ff}(\boldsymbol{\theta}, \Theta) \end{bmatrix} \in \mathbb{R}^{(d_{\mathbf{y}}+d_f+d_g)\times N(d_{\mathbf{y}}+d_f+d_g)},$$

$$\tilde{K}(\Theta, \Theta) = \begin{bmatrix} K_{uu}(\Theta, \Theta) & K_{ug}(\Theta, \Theta) & K_{uf}(\Theta, \Theta) \\ K_{ug}^{\mathrm{T}}(\Theta, \Theta) & K_{gg}(\Theta, \Theta) & K_{gf}(\Theta, \Theta) \\ K_{uf}^{\mathrm{T}}(\Theta, \Theta) & K_{gf}^{T}(\Theta, \Theta) & K_{ff}(\Theta, \Theta) \end{bmatrix} \in \mathbb{R}^{N(d_{\mathbf{y}}+d_f+d_g)\times N(d_{\mathbf{y}}+d_f+d_g)},$$

$$\mathbf{g}(\Theta) = \begin{bmatrix} \mathcal{G}_X(\Theta) \\ g(\Theta, X_g) \\ f(\Theta, X_f) \end{bmatrix} \in \mathbb{R}^{N(d_{\mathbf{y}}+d_f+d_g)},$$

and

$$K_{uu}(\Theta, \Theta) = \{k_p(\boldsymbol{\theta}^i, \boldsymbol{\theta}^j)K_s(X, X)\} \in \mathbb{R}^{Nd_{\mathbf{y}}\times Nd_{\mathbf{y}}}, \quad \text{similarly} \quad K_{uu}(\boldsymbol{\theta}, \Theta) \in \mathbb{R}^{d_{\mathbf{y}}\times Nd_{\mathbf{y}}},$$
$$K_{ug}(\Theta, \Theta) = \{k_p(\boldsymbol{\theta}^i, \boldsymbol{\theta}^j)\mathcal{B}K_s(X, X_g)\} \in \mathbb{R}^{Nd_{\mathbf{y}}\times Nd_g}, \quad \text{similarly} \quad K_{ug}(\boldsymbol{\theta}, \Theta) \in \mathbb{R}^{d_{\mathbf{y}}\times Nd_g},$$
$$K_{uf}(\Theta, \Theta) = \{k_p(\boldsymbol{\theta}^i, \boldsymbol{\theta}^j)\mathcal{L}^{\boldsymbol{\theta}^j}K_s(X, X_f)\} \in \mathbb{R}^{Nd_{\mathbf{y}}\times Nd_f}, \quad \text{similarly} \quad K_{uf}(\boldsymbol{\theta}, \Theta) \in \mathbb{R}^{d_{\mathbf{y}}\times Nd_f},$$
$$K_{gg}(\Theta, \Theta) = \{k_p(\boldsymbol{\theta}^i, \boldsymbol{\theta}^j)\mathcal{B}\mathcal{B}K_s(X_g, X_g)\} \in \mathbb{R}^{Nd_g\times Nd_g}, \quad \text{similarly} \quad K_{gg}(\boldsymbol{\theta}, \Theta) \in \mathbb{R}^{d_g\times Nd_g},$$
$$K_{gf}(\Theta, \Theta) = \{k_p(\boldsymbol{\theta}^i, \boldsymbol{\theta}^j)\mathcal{B}\mathcal{L}^{\boldsymbol{\theta}^j}K_s(X_g, X_f)\} \in \mathbb{R}^{Nd_g\times Nd_f}, \quad \text{similarly} \quad K_{gf}(\boldsymbol{\theta}, \Theta) \in \mathbb{R}^{d_g\times Nd_f},$$
$$K_{ff}(\Theta, \Theta) = \{k_p(\boldsymbol{\theta}^i, \boldsymbol{\theta}^j)\mathcal{L}^{\boldsymbol{\theta}^i}\mathcal{L}^{\boldsymbol{\theta}^j}K_s(X_f, X_f)\} \in \mathbb{R}^{Nd_f\times Nd_f}, \quad \text{similarly} \quad K_{ff}(\boldsymbol{\theta}, \Theta) \in \mathbb{R}^{d_f\times Nd_f},$$
$$g(\Theta, X_g) = \{g(\boldsymbol{\theta}^i, X_g)\} \in \mathbb{R}^{Nd_g},$$
$$f(\Theta, X_f) = \{f(\boldsymbol{\theta}^i, X_f)\} \in \mathbb{R}^{Nd_f}.$$

The marginal posterior distribution on $\mathcal{G}_X(\boldsymbol{\theta})$ can then be extracted from the above joint posterior by taking the first $d_{\mathbf{y}}$ rows of $\mathbf{m}_N^{\mathbf{g}}$ and the first $d_{\mathbf{y}}$ rows and columns of $K_N^{\mathbf{g}}$, which gives

$$\mathcal{G}_X(\boldsymbol{\theta})|\mathcal{G}_X(\Theta), g(\Theta, X_g), f(\Theta, X_f) \sim \mathrm{GP}(\mathbf{m}_{N,X_f,X_g}^{\mathcal{G}_X}(\boldsymbol{\theta}), K_{N,X_f,X_g}(\boldsymbol{\theta}, \boldsymbol{\theta}')), \quad (19)$$

where

$$\mathbf{m}_{N,X_f,X_g}^{\mathcal{G}_X}(\boldsymbol{\theta}) = \begin{bmatrix} K_{uu}(\boldsymbol{\theta}, \Theta) & K_{ug}(\boldsymbol{\theta}, \Theta) & K_{uf}(\boldsymbol{\theta}, \Theta) \end{bmatrix} \tilde{K}(\Theta, \Theta)^{-1}\mathbf{g}(\Theta),$$

$$K_{N,X_f,X_g}(\boldsymbol{\theta}, \boldsymbol{\theta}') = K(\boldsymbol{\theta}, \boldsymbol{\theta}') - \begin{bmatrix} K_{uu}(\boldsymbol{\theta}, \Theta) & K_{ug}(\boldsymbol{\theta}, \Theta) & K_{uf}(\boldsymbol{\theta}, \Theta) \end{bmatrix} \tilde{K}(\Theta, \Theta)^{-1} \begin{bmatrix} K_{uu}(\boldsymbol{\theta}', \Theta) \\ K_{ug}(\boldsymbol{\theta}', \Theta) \\ K_{uf}(\boldsymbol{\theta}', \Theta) \end{bmatrix}.$$

Compared to the spatially correlated posterior in (16), note that in (19) we are updating our prior on $\mathcal{G}_X(\boldsymbol{\theta})$ using the observations $g(\Theta, X_g)$ and $f(\Theta, X_f)$ as well as $\mathcal{G}_X(\Theta)$. Since the g and f are assumed known, these additional observations are cheap to obtain. It is also possible to extend the methodology to condition on training data $g(\Theta_g, X_g)$ and $f(\Theta_f, X_f)$, for point sets Θ_g and Θ_f different to Θ, and this has been found to be beneficial in some of the numerical experiments (see Section 4).

Note that when emulating the potential Φ instead of the forward map $\mathcal{G}_X$, we are emulating a scalar-valued function. Since Φ is a non-linear function of $\mathcal{G}_X$, it is not possible to extend the ideas of spatial correlation presented in this section to emulating Φ, and in particular, it is not possible to construct a PDE-informed emulator in the same way.

3.1.1 Computational implementation

We now have three different approaches for emulating the forward map and defining the correlation between its components. We will refer to these as the independent, spatially correlated, and PDE-constrained model, respectively. Each of them can be combined with the mean-based or the marginal approximation of the posterior. We note here that for the computational implementation of the spatially correlated model, the introduction of the correlation matrix does not change the predictive mean of the Gaussian process, it only affects the predictive covariance (see Theorem 1 below). Since the spatial correlation matrix is independent of $\boldsymbol{\theta}$, the covariance matrix between two sets of parameters Θ_1 and Θ_2 can be computed by the Kronecker product, that is,

$$\underbrace{K(\Theta_1, \Theta_2)}_{N_1 d_{\mathbf{y}} \times N_2 d_{\mathbf{y}}} = \underbrace{K_p(\Theta_1, \Theta_2)}_{(N_1 \times N_2)} \otimes \underbrace{K_s(X, X)}_{(d_{\mathbf{y}} \times d_{\mathbf{y}})}. \tag{20}$$

Hence, assuming a spatial correlation of the type (14) only affects approximate posteriors that take into account the uncertainty of the emulator.

Theorem 1. *Consider two Gaussian processes $\mathbf{g}_0(\boldsymbol{\theta}) \sim GP(\mathbf{m}(\boldsymbol{\theta}), k_p(\boldsymbol{\theta}, \boldsymbol{\theta}')I_{d_{\mathbf{y}}})$ and $\mathbf{g}_{0,s}(\boldsymbol{\theta}) \sim GP(\mathbf{m}(\boldsymbol{\theta}), k_p(\boldsymbol{\theta}, \boldsymbol{\theta}')K_s(X, X))$, where $K_s(X, X)$ is the covariance matrix on the set of spatial points $X = \{\mathbf{x}_i\}_{i=1}^{d_{\mathbf{y}}}$ and $k_p(\boldsymbol{\theta}, \boldsymbol{\theta}')$ is scalar-valued. Conditioning both Gaussian processes on a set of training points $\mathbf{g}(\Theta) = \{\mathbf{g}(\boldsymbol{\theta}_i)\}_{i=1}^N$, denote the corresponding conditional Gaussian processes by $\mathbf{g}^N(\boldsymbol{\theta}) \sim GP(\mathbf{m}_N^{\mathbf{g}}(\boldsymbol{\theta}), K_N(\boldsymbol{\theta}, \boldsymbol{\theta}'))$ and $\mathbf{g}_s^N(\boldsymbol{\theta}) \sim GP(\mathbf{m}_{N,s}^{\mathbf{g}}(\boldsymbol{\theta}), K_{N,s}(\boldsymbol{\theta}, \boldsymbol{\theta}'))$, respectively. Then we have,*

$$\begin{aligned} \mathbf{m}_{N,s}^{\mathbf{g}}(\boldsymbol{\theta}) &= \mathbf{m}_N^{\mathbf{g}}(\boldsymbol{\theta}), \\ K_N(\boldsymbol{\theta}, \boldsymbol{\theta}') = k_{N,p}(\boldsymbol{\theta}, \boldsymbol{\theta}')I_{d_{\mathbf{y}}}, &\quad \text{and} \quad K_{N,s}(\boldsymbol{\theta}, \boldsymbol{\theta}') = k_{N,p}(\boldsymbol{\theta}, \boldsymbol{\theta}')K_s(X, X), \end{aligned}$$

where $k_{N,p}(\boldsymbol{\theta}, \boldsymbol{\theta}')$ is scalar-valued.

Proof. We now prove the expression for the predictive mean. Let $k_p(\boldsymbol{\theta}, \Theta) := [k_p(\boldsymbol{\theta}, \boldsymbol{\theta}^1); \ldots; k_p(\boldsymbol{\theta}, \boldsymbol{\theta}^N)] \in \mathbb{R}^{1 \times d_{\mathbf{y}}}$, and denote by $K_p(\Theta, \Theta) \in \mathbb{R}^{d_{\mathbf{y}} \times d_{\mathbf{y}}}$ the matrix with entries $(K_p(\Theta, \Theta))_{i,j} = k_p(\boldsymbol{\theta}^i, \boldsymbol{\theta}^j)$. Then by (8) we have

$$\begin{aligned} &\mathbf{m}_{N,s}^{\mathbf{g}}(\boldsymbol{\theta}) \\ &= \mathbf{m}(\boldsymbol{\theta}) + \left(k_p(\boldsymbol{\theta}, \Theta) \otimes K_s(X, X)\right)^T \left(K_p(\Theta, \Theta) \otimes K_s(X, X)\right)^{-1} \left(\mathbf{g}(\Theta) - \mathbf{m}(\Theta)\right), \end{aligned}$$

where $\otimes$ denotes the Kronecker product. Using properties of products and inverses of Kronecker products and the fact that $K_s(X, X)$ is symmetric positive

definite, we then have

$$
\begin{aligned}
&\mathbf{m}^{\mathbf{g}}_{N,s}(\boldsymbol{\theta}) \\
&= \mathbf{m}(\boldsymbol{\theta}) + \left(k_p(\boldsymbol{\theta},\Theta)^T \otimes K_s(X,X)^T\right)\left(K_p(\Theta,\Theta)^{-1} \otimes K_s(X,X)^{-1}\right)(\mathbf{g}(\Theta) - \mathbf{m}(\Theta)) \\
&= \mathbf{m}(\boldsymbol{\theta}) + \left(k_p(\boldsymbol{\theta},\Theta)^T K_p(\Theta,\Theta)^{-1} \otimes K_s(X,X)^T K_s(X,X)^{-1}\right)(\mathbf{g}(\Theta) - \mathbf{m}(\Theta)) \\
&= \mathbf{m}(\boldsymbol{\theta}) + \left(k_p(\boldsymbol{\theta},\Theta)^T K_p(\Theta,\Theta)^{-1} \otimes I_{d_{\mathbf{y}}}\right)(\mathbf{g}(\Theta) - \mathbf{m}(\Theta)) \\
&= \mathbf{m}^{\mathbf{g}}_{N}(\boldsymbol{\theta}).
\end{aligned}
$$

The relationship between $K_{N,s}(\boldsymbol{\theta},\boldsymbol{\theta}')$ and $K_N(\boldsymbol{\theta},\boldsymbol{\theta}')$ can be shown in a similar way, using (9).

□

For the PDE-constrained model, since the covariance functions related to f are obtained by applying the differential operator, the spatially correlated matrix in the joint prior (17) also depends explicitly on the parameters $\boldsymbol{\theta}$. Therefore, its covariance matrix cannot be written in a Kronecker product structure as in (20) and Theorem 1 does not apply. Thus, incorporating the PDE constraints into the model also affects the predictive mean and hence the mean-based posterior is also changed.

3.2 MCMC algorithms

To extract information from the posterior, MCMC algorithms are powerful and popular tools [26, 3]. In this work, we will consider the Metropolis-Adjusted Langevin Algorithm (MALA) [27], which is a type of MCMC algorithm that uses gradient information to accelerate the convergence of the sampling chain. Central to the idea of MALA, and any gradient-based sampling method in fact, is the overdamped Langevin stochastic differential equation (SDE):

$$
d\boldsymbol{\theta} = \nabla \log \pi(\boldsymbol{\theta}|\mathbf{y})dt + \sqrt{2}dW, \tag{21}
$$

where W is a standard $d_{\boldsymbol{\theta}}$-dimensional Brownian motion. Under mild conditions on the posterior π [26], (21) is ergodic and has π as its stationary distribution, so that the probability density function of $\boldsymbol{\theta}(t)$ tends to π as $t \to \infty$.

In practice, the dynamics (21) is discretised with a simple Euler-Maruyama method with a time step γ.

$$
\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n + \gamma \nabla \log \pi(\boldsymbol{\theta}|\mathbf{y}) + \sqrt{2\gamma}\xi_n, \tag{22}
$$

with $\xi_n \sim \mathcal{N}(0,1)$. Assuming that the dynamics of (22) remain ergodic the corresponding numerical invariant measure would not necessarily coincide with the posterior. To alleviate this bias, one needs to incorporate an accept-reject mechanism. This gives rise to MALA as descibed in Algorithm 1.

An advantage of using the Gaussian process emulator in the posterior is that, assuming the prior is differentiable, $\nabla \log \pi^N(\boldsymbol{\theta}|\mathbf{y})$ can be computed analytically for the mean-based and marginal approximations introduced in Section

Algorithm 1 Metropolis-Adjusted Langevin Algorithm

Require: initial value $\boldsymbol{\theta}_0$, number of samples N, time-step γ, posterior $\pi(\boldsymbol{\theta}|\mathbf{y})$

while $n < N$ **do**

1. Generate $\xi_n \sim \mathcal{N}(0,1)$.
2. Generate a candidate

$$\boldsymbol{\theta}' = \boldsymbol{\theta}_n + \gamma \nabla \log \pi(\boldsymbol{\theta}_n|\mathbf{y}) + \sqrt{2\gamma}\xi_n.$$

3. Compute the acceptance rate

$$\alpha_n := \max\left(1, \frac{\pi(\boldsymbol{\theta}'|\mathbf{y})q(\boldsymbol{\theta}_n|\boldsymbol{\theta}')}{\pi(\boldsymbol{\theta}_n|\mathbf{y})q(\boldsymbol{\theta}'|\boldsymbol{\theta}_n)}\right),$$

where $q(\boldsymbol{\theta}|\tilde{\boldsymbol{\theta}}) \propto \exp\left(-\frac{1}{4\gamma}\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}} - \gamma\nabla \log \pi(\tilde{\boldsymbol{\theta}}|\mathbf{y})\|^2\right)$.

4. Generate $r \sim U[0,1]$. If $r > \alpha_n$, set $\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}'$; otherwise $\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n$.

end while

2.3, which enables us to easily implement the MALA algorithm. Note that in contrast since the true posterior involves (analytical or numerical) solution u to the PDE (1a)-(1b), it is usually impossible to compute these gradients analytically. The following Lemma gives us the gradient of the different approximate posteriors

Lemma 2. *Given a Gaussian process $\mathcal{G}_X^N \sim GP(\mathbf{m}_N^{\mathcal{G}_X}(\boldsymbol{\theta}), K_N(\boldsymbol{\theta},\boldsymbol{\theta}))$ emulating the forward map $\mathcal{G}_X$ with data $\mathcal{G}_X(\Theta)$, then the gradient of the mean-based approximation of the posterior*

$$\nabla \log(\pi_{\text{mean}}^{N,\mathcal{G}_X}(\boldsymbol{\theta}|\mathbf{y})) = -\frac{1}{\sigma_\eta^2}\nabla \mathbf{m}_N^{\mathcal{G}_X}(\boldsymbol{\theta})^T(\mathbf{m}_N^{\mathcal{G}_X}(\boldsymbol{\theta}) - \mathbf{y}) + \nabla \log \pi_0(\boldsymbol{\theta}),$$

and the gradient of the marginal approximation of the posterior

$$\begin{aligned}\nabla \log(\pi_{\text{marginal}}^{N,\mathcal{G}_X}(\boldsymbol{\theta}|\mathbf{y})) = &-\nabla \mathbf{m}_N^{\mathcal{G}_X}(\boldsymbol{\theta})^T(K_N(\boldsymbol{\theta},\boldsymbol{\theta}) + \Gamma_\eta)^{-1}(\mathbf{m}_N^{\mathcal{G}_X}(\boldsymbol{\theta}) - \mathbf{y}) \\ &- \frac{1}{2}(\mathbf{m}_N^{\mathcal{G}_X}(\boldsymbol{\theta}) - \mathbf{y})^T\nabla\left((K_N(\boldsymbol{\theta},\boldsymbol{\theta}) + \Gamma_\eta)^{-1}\right)(\mathbf{m}_N^{\mathcal{G}_X}(\boldsymbol{\theta}) - \mathbf{y}) \\ &- \frac{1}{2}\left(\text{Tr}\left((K_N(\boldsymbol{\theta},\boldsymbol{\theta}) + \Gamma_\eta)^{-1}\right)\nabla(K_N(\boldsymbol{\theta},\boldsymbol{\theta}))\right) + \nabla \log \pi_0(\boldsymbol{\theta}),\end{aligned}$$

where

$$\nabla\left((K_N(\boldsymbol{\theta},\boldsymbol{\theta}) + \Gamma_\eta)^{-1}\right) = -(K_N(\boldsymbol{\theta},\boldsymbol{\theta}) + \Gamma_\eta)^{-1}\nabla\left(K_N(\boldsymbol{\theta},\boldsymbol{\theta})\right)(K_N(\boldsymbol{\theta},\boldsymbol{\theta}) + \Gamma_\eta)^{-1},$$

and

$$\nabla K_N(\boldsymbol{\theta},\boldsymbol{\theta}) = 2\nabla K(\boldsymbol{\theta},\Theta)K(\Theta,\Theta)^{-1}K(\Theta,\boldsymbol{\theta})$$

4 Numerical experiments

We now discuss a number of different numerical experiments related to inverse problems for the PDE (1a)-(1b) in various set-ups in terms of the number of spatial and parameter dimensions as well as for different types of forward models. In cases where the PDE solution is not available in closed form, we use the finite element software Firedrake [25] to obtain the "true" solution. Furthermore, in all our numerical experiments we replace the uniform prior by a smooth approximation given by the $\lambda-$Moreau-Yoshida envelope [2] with $\lambda = 10^{-3}$. To further clarify the notation we use in our numerical experiment, we introduce part of them again in the following table (see Table 2).

Symbol	Description
$\mathcal{G}_X(\Theta)$	Training data set: point-wise evaluation of the PDE solution $u(\boldsymbol{\theta}, \mathbf{x})$ for $\mathbf{x} \in X = \{\mathbf{x}_i\}_{i=1}^{d_\mathbf{y}}$, $\boldsymbol{\theta} \in \Theta = \{\boldsymbol{\theta}^i\}_{i=1}^{N}$
$g(\Theta_g, X_g)$	Additional training data for boundary condition: point-wise evaluation of the function $g(\boldsymbol{\theta}, \mathbf{x})$ for $\mathbf{x} \in X_g = \{\mathbf{x}_i\}_{i=1}^{d_g}$, $\boldsymbol{\theta} \in \Theta = \{\boldsymbol{\theta}^i\}_{i=1}^{N_g}$
$f(\Theta_f, X_f)$	Additional training data for the source function: point-wise evaluation of the function $f(\boldsymbol{\theta}, \mathbf{x})$ for $\mathbf{x} \in X = \{\mathbf{x}_i\}_{i=1}^{d_f}$, $\boldsymbol{\theta} \in \Theta = \{\boldsymbol{\theta}^i\}_{i=1}^{N_f}$
$\bar{N}$	In practice, we use $N_g = N_f = \bar{N}$

Table 2: Symbols and notations used in numerical experiments.

4.1 Examples in one spatial dimension

4.1.1 Constant diffusion coefficient

We consider the following PDE in one spatial dimension

$$\begin{aligned} -\frac{\mathrm{d}}{\mathrm{d}x}\left(e^{\theta}\frac{\mathrm{d}u(x)}{\mathrm{d}x}\right) &= 1, \qquad x \in (0,1),\, \theta \in [-1,1], \\ u(0) = 0, u(1) &= 0. \end{aligned} \tag{23}$$

In this case the dimension of the parameter space is $d_{\boldsymbol{\theta}} = 1$, and the solution is available in closed form. More precisely, we have

$$u(x) = \frac{(x - x^2)}{2e^{\theta}}.$$

Given this explicit solution and the low dimension of the parameter space, it is possible to calculate the true and approximate posteriors without having to resort to Markov Chain Monte Carlo sampling. We now generate our observations $\mathbf{y}$ according to equation (2) for the value of $\boldsymbol{\theta}^\dagger = 0.314$ for a varying number of spatial points $d_\mathbf{y}$ (equally spaced in $[0,1]$) and for noise level $\sigma_\eta^2 = 10^{-5}$. As

we can see in Figure 1 as we increase $d_{\mathbf{y}}$ the true posterior $\pi(\theta|\mathbf{y})$ tends to get more and more concentrated around the value of $\boldsymbol{\theta}^{\dagger}$ which is consistent with what the theory would predict by a Bernstein-von-Mises theorem (see e.g. [12] for related results).

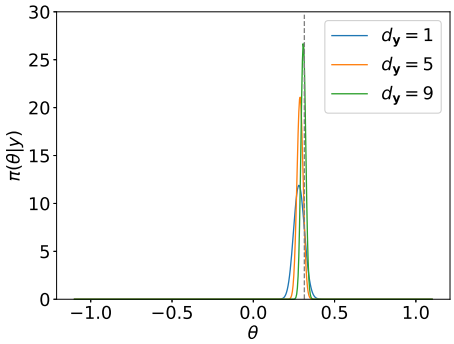
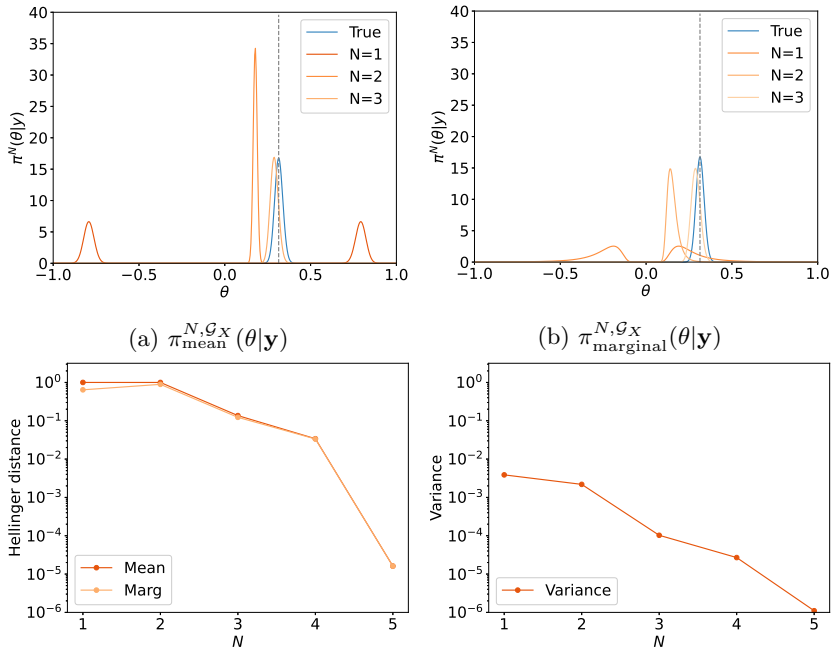


Figure 1: True posterior with different $d_{\mathbf{y}}$

We now turn our attention to the different approximate posteriors discussed in Section 2.3 obtained for different Gaussian priors (independent, spatially correlated, and PDE-constrained).

Baseline model: In the case of the simplest emulator with independent entries, we illustrate in Figure 2, how the mean-based posterior $\pi^{N,\mathcal{G}_X}_{\text{mean}}(\theta|\mathbf{y})$ and the marginal posterior $\pi^{N,\mathcal{G}_X}_{\text{marginal}}(\theta|\mathbf{y})$ behave as a function of the number of training points N (here $d_{\mathbf{y}} = 5$). The location of the training points is chosen from the Halton sequence [21]. Now, when comparing Figure 2(a) and 2(b) we see that the marginal posterior is more spread than the mean-based posterior. This is due to the variance inflation associated with the marginal posterior which reflects better the uncertainty of the emulator. For example, in the case $N = 1$ the mean-based posterior has negligible posterior probability mass near $\boldsymbol{\theta}^{\dagger}$, while due to the variance inflation this is not the case for the marginal-based posterior. Furthermore, in Figure 2(c) we plot the Hellinger distance between the approximate posteriors and the true posterior as a function of the number of training points N. As we can see the error for the marginal-based posterior is smaller than the error for the mean-based posterior for small N while the two errors behave in the same way as N increases. This can be further understood by Figure 2(d) where we plot the average variance of our emulator for different values of N and see that the value of N for which the error between the two posteriors is equal corresponds to the value of N for which the average variance of the emulator is of the same order as the variance of the observational noise σ^2_{η}.

Spatially correlated model: As discussed in Section 3.1 the introduction of spatial correlation doesn't change the predictive mean of the Gaussian pro-



(a) $\pi^{N,\mathcal{G}_X}_{\text{mean}}(\theta|\mathbf{y})$

(b) $\pi^{N,\mathcal{G}_X}_{\text{marginal}}(\theta|\mathbf{y})$

(c) Hellinger distance

(d) Average variance of the emulator

Figure 2: (2a) Baseline model mean-based posterior with different N. (2b) Baseline model marginal posterior with different N. (2c) Hellinger distance between approximated posteriors and true posteriors when N increases. (2d) Average predictive variance of the Gaussian process emulator as N increases. $\mathcal{G}_X$ is the discretised solution u in (23).

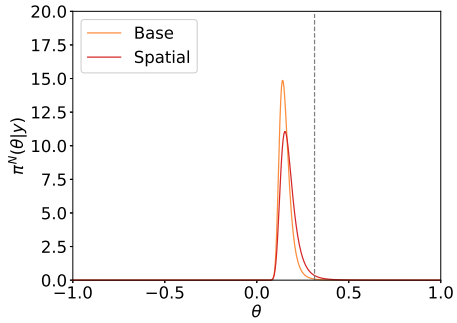
cesses. We hence now compare in Figure 3 the two different marginal posteriors $\pi^{N,\mathcal{G}_X}_{\text{marginal}}$, $\pi^{N,\mathcal{G}_X,s}_{\text{marginal}}$, where the latter includes spatial correlation. We again choose $d_{\mathbf{y}} = 5$. In particular, as we can see in Figure 3(a) (here $N = 2$), introducing spatial correlation seems to improve the accuracy of the approximate posterior and place more mass near $\theta^\dagger$. The fact that the spatially correlated model has an increased variance at $N = 2$ (see Figure 3c) leads to similar behavior as in Figure 2 with $\pi^{N,\mathcal{G}_X,s}_{\text{marginal}}$ being more spread than $\pi^{N,\mathcal{G}_X}_{\text{marginal}}$. Furthermore, as we can see in Figure 3(b) as we increase the number of training points for our Gaussian process the Hellinger distance between the true posterior and $\pi^{N,\mathcal{G}_X,s}_{\text{marginal}}$ is smaller than the one of the baseline model.

PDE-constrained model: We now compare the behaviour of the PDE-constrained model with the other two models, both for mean-based approximate posterior, as well as for marginal posterior (again here $d_{\mathbf{y}} = 5$). In particular, as we can see in Figures 4(a) and 4(b) for $N = 2$, $\pi^{N,\mathcal{G}_X,\text{PDE}}_{\text{mean}}$ and $\pi^{N,\mathcal{G}_X,\text{PDE}}_{\text{marginal}}$ are indistinguishable from the true posterior when using $\bar{N} = 10$, $d_f = 5$ showing much better approximation properties than the other two models. This is consistent with what we observe in terms of Hellinger distance, since both $\pi^{N,\mathcal{G}_X,\text{PDE}}_{\text{mean}}$ and $\pi^{N,\mathcal{G}_X,\text{PDE}}_{\text{marginal}}$ have similar errors over a different range of values for N_f. It is also worth noting that when comparing with the Hellinger distance from Figures Figures 2(c) and 3(c) we see that the PDE-based model achieves the same order of error with only using half of the training points ($N = 2$ instead of $N = 4$). Furthermore, as we can see in Figure 4(d) the average variance of the PDE-constrained emulator converges to zero very fast as the number of extra training points for f increases, implying that at least in this simple example adding the PDE knowledge leads to an extremely good approximation of the forward map.

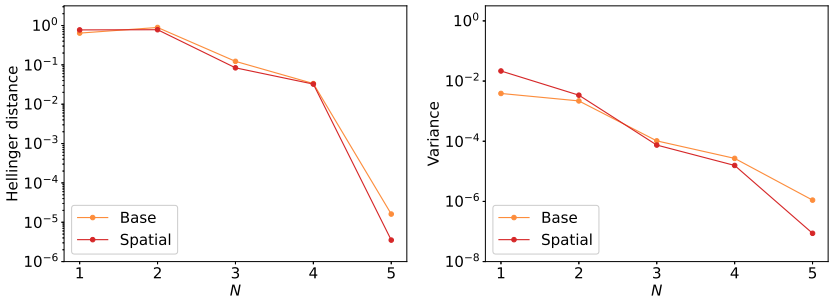
4.1.2 Two dimensional piece-wise constant diffusion coefficient

We now consider a slightly more general problem than (23). In particular, we consider the same elliptic equation as in (23) but use a 2-dimensional piece-wise constant diffusion coefficient. In particular, we now have the following equation

$$\begin{aligned} -\frac{\mathrm{d}}{\mathrm{d}x}(\exp(\kappa(x,\boldsymbol{\theta}))\frac{\mathrm{d}}{\mathrm{d}x}u(x)) &= 4x, \qquad x \in (0,1),\, \boldsymbol{\theta} \in [-1,1], \\ u(0) &= 0, \\ u(1) &= 2, \end{aligned} \tag{24}$$

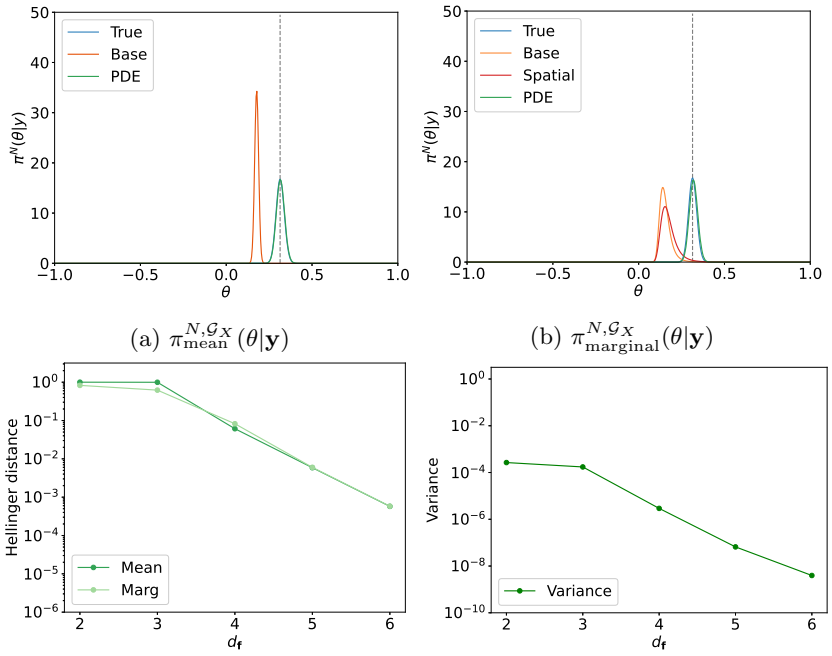


(a) $\pi^{N,\mathcal{G}_X}_{\text{marginal}}(\theta|\mathbf{y})$, $\pi^{N,\mathcal{G}_X,s}_{\text{marginal}}(\theta|\mathbf{y})$ $(N = 2)$



(b) Hellinger distance (c) Average variance of the emulator

Figure 3: (3a) Baseline and spatially correlated model marginal posterior for $N = 2$. (3b) Hellinger distance between approximated posteriors and true posterior as N increases. (3c) Average predictive variance of the Gaussian process emulator as N increases. $\mathcal{G}_X$ is the discretised solution u in (23).



(a) $\pi^{N,\mathcal{G}_X}_{\text{mean}}(\theta|\mathbf{y})$

(b) $\pi^{N,\mathcal{G}_X}_{\text{marginal}}(\theta|\mathbf{y})$

(c) Hellinger distance

(d) Average variance of the emulator

Figure 4: Comparison of different models when $N = 2$, for PDE model $d_f = 5$. (4a) Mean-based posteriors (4b) Marginal posteriors (4c) Hellinger distance between approximated posteriors and true posterior as d_f increases. (4d) Average predictive variance of emulator as d_f increases. $\mathcal{G}_X$ is the discretised solution u in (23).

where κ is defined as piece-wise constant over four equally spaced intervals. More precisely, we consider

$$\kappa(x, \boldsymbol{\theta}) = \begin{cases} 0, & \text{for } x \in [0, \frac{1}{4}) \\ \theta_1, & \text{for } x \in [\frac{1}{4}, \frac{1}{2}) \\ \theta_2, & \text{for } x \in [\frac{1}{2}, \frac{3}{4}) \\ 1 & \text{for } x \in [\frac{3}{4}, 1] \end{cases} \tag{25}$$

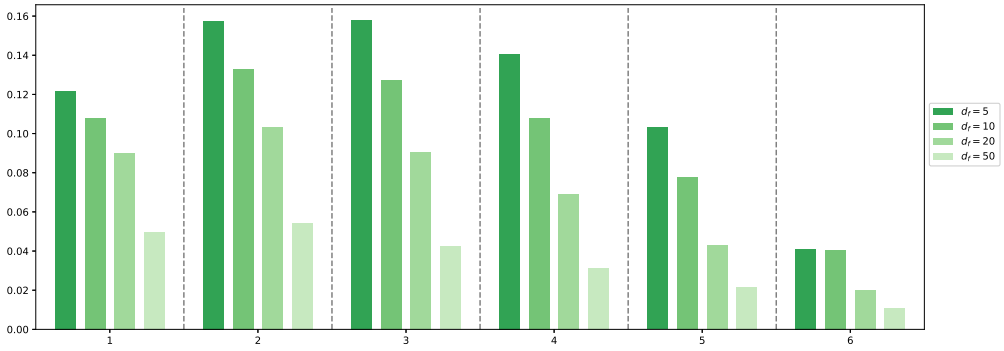
Unlike equation (23), it is not possible to obtain an analytic solution for (24) so we use instead Firedrake to obtain its solution.

For the PDE constrained model, we first test the effectiveness of additional training data $g(\Theta_g, X_g)$ and $f(\Theta_f, X_f)$. We let the size of point set Θ_g and Θ_g to be the same and denoted by $\bar{N}$. In Figure 5, we test the impact of d_f and $\bar{N}$ to the accuracy of the PDE constrained emulator. We use fixed same hyperparameters for all the models and $d_g = 2$. We see that as d_f increases the accuracy of emulators gradually increases. While for $\bar{N}$, we see that a certain amount of additional point can improve the accuracy, but including more points cannot make further improvement.

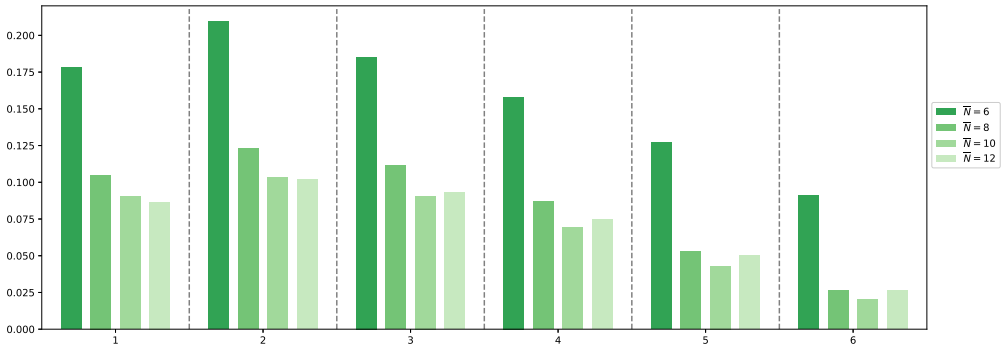
Throughout this numerical experiment, we take the prior of the parameters to be the uniform distribution on $[-1, 1]^2$, and we generate our data $\mathbf{y}$ according to equation (2) for the value $\boldsymbol{\theta}^\dagger = [0.098, 0.430]$ for $d_{\mathbf{y}} = 6$ (equally spaced points in [0,1]) and for noise level $\sigma_\eta^2 = 10^{-4}$. For the baseline and spatially correlated model, we have used $N = 4$ training points (chosen to be the first 4 points in the Halton sequence), while additionally for the PDE-constrained model, we have used $\bar{N} = 10$ (chosen to be the following 10 points in the Halton sequence) and $d_f = 20$. For the covariance kernels, we choose k_p to be the squared exponential kernel and k_s to be the Matèrn kernel with $\nu = \frac{5}{2}$.

Unlike (23) we now do not perform exact integration but use the MALA algorithm to obtain our samples. In particular, for all our approximate posteriors we have used 10^6 samples. In addition, since in this case, we do not have an analytic expression for the solution, we do not have direct access to the true posterior. We circumvent this problem by considering the results obtained by a mean-based approximation with the baseline model for $N = 10^2$ training points as the ground truth.

As we can see in Figures 6(a)-(c) all the mean-based posteriors are failing to put significant posterior mass near the true parameter value $\boldsymbol{\theta}^\dagger$. The situation improves when the uncertainty of the emulator is taken into account as we can see for the marginal-based posteriors. Out of the three different models, the PDE-constrained one seems to be performing best since it is placing the most posterior mass around the true value $\boldsymbol{\theta}^\dagger$. This is further illustrated in Figure 7 where we plot the θ_1 and θ_2 marginals for all the mean-based posterior approximations $\pi_{\text{mean}}^{N,\mathcal{G}_X}$, $\pi_{\text{mean}}^{N,\mathcal{G}_X,s}$, $\pi_{\text{mean}}^{N,\mathcal{G}_X,\text{PDE}}$ and the marginal-based posterior approximations $\pi_{\text{marginal}}^{N,\mathcal{G}_X}$, $\pi_{\text{marginal}}^{N,\mathcal{G}_X,s}$, $\pi_{\text{marginal}}^{N,\mathcal{G}_X,\text{PDE}}$. Note that the marginal plot

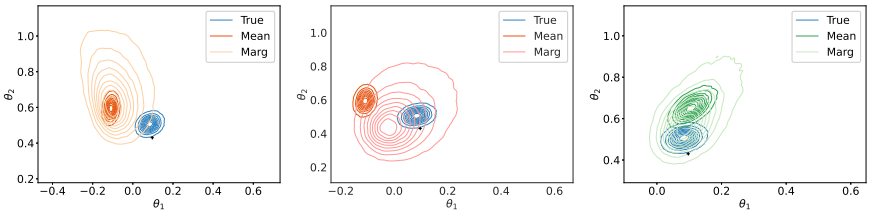


(a)



(b)

Figure 5: Error between the predictive mean of PDE constrained emulators and the ground truth at observation points ($\boldsymbol{\theta} = \boldsymbol{\theta}^\dagger$) for different (a) d_f ($\bar{N} = 10$) (b) $\bar{N}$ ($d_f = 20$)



(a) Baseline (b) Spatially correlated (c) PDE-constrained

Figure 6: Contour plots of the approximate mean-based and inflated-based posteriors. (a) baseline model (b) spatially correlated (c) PDE-constrained. ” + ” denotes $\boldsymbol{\theta}^\dagger$. $\mathcal{G}_X$ is the discretised solution u in (24).

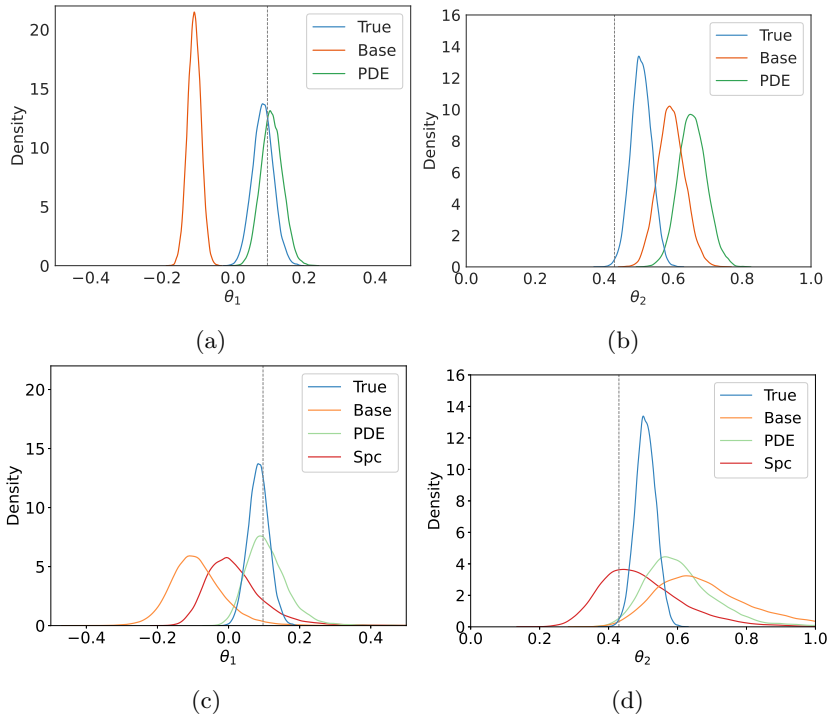


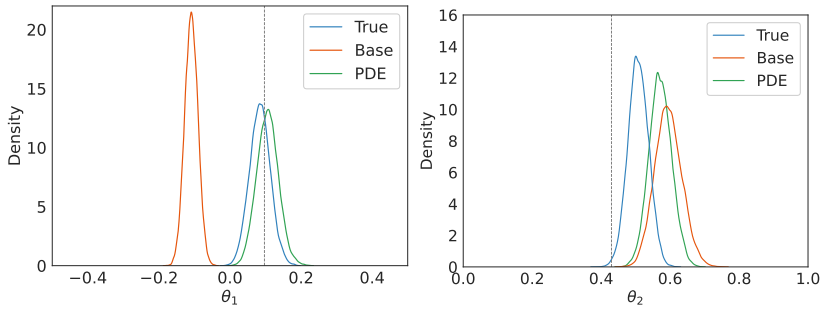
Figure 7: Comparison of different models' marginal distribution when $N = 4$, for PDE model $d_f = 20$ and $\bar{N} = 20$. (a) Mean-based approximation θ_1. (b) Mean-based approximation θ_2. (c) Marginal approximation θ_1. (d) Marginal approximation θ_2. $\mathcal{G}_X$ is the discretised solution u in (24) with diffusion coefficient (25).

could be misleading the overall performance of the approximations, for example in Figure 7b the baseline model seems to be better than the PDE-constrained model, but from the Figure 6 we know that is not true. When we increase d_f from 20 to 50, the accuracy of approximation improves. We see that in Figure 8, the marginal plot of the mean-based approximate

4.1.3 Integral observation operator

We now investigate the proposed method with a different form of observation operator. In terms of the PDE problem, we study again (24). However, instead of point-wise observations $\mathcal{G}_X(\boldsymbol{\theta}) = \{u(x_j;\boldsymbol{\theta})\}_{j=1}^{d_{\mathbf{y}}}$ as in (2), we observe local averages $\mathcal{G}_X(\boldsymbol{\theta}) = \{\int_{a_j}^{b_j} u(x;\boldsymbol{\theta})dx\}_{j=1}^{d_{\mathbf{y}}}$ for non-overlapping intervals $[a_j, b_j \subset [0,1]$.

For the inverse problem setting, we have $\boldsymbol{\theta}^\dagger = [0.098, 0.430]$ which is the same as before, $d_{\mathbf{y}} = 16$ (equally spaced sub-intervals of $[0,1]$) and $\sigma_\eta^2 = 10^{-6}$.



(a) (b)

Figure 8: Comparison of different models' marginal distribution when $N = 4$, for PDE model $d_f = 50$ and $\bar{N} = 20$. (a) Mean-based approximation θ_1. (b) Mean-based approximation θ_2. (c) Marginal approximation θ_1. (d) Marginal approximation θ_2. $\mathcal{G}_X$ is the discretised solution u in (24) with diffusion coefficient (25).

We again do not conduct precise integration as in (23), but use MALA algorithm to obtain our samples. We utilize 10^6 samples for all our approximate posterior. We treat the sampling results obtained by a mean-based approximation with the baseline model for $N = 10^2$ training points as the ground truth. In Figure 9, we plot again the θ_1 and θ_2 marginals for all the mean-based posterior approximations and the marginal posterior approximations. The result is similar to the previous example that the PDE-constrained model performs better than the other two models.

4.1.4 Parametric expansion for the diffusion coefficient

In this example, we study again (24), but this time instead of working with a piecewise constant diffusion coefficient we assume that the diffusion coefficient satisfies the following parametric expansion

$$\kappa(\boldsymbol{\theta}, x) = \exp\left(\sum_{n=1}^{2} \sqrt{a_n}\theta_n b_n(x)\right) \tag{26}$$

where $a_n = \frac{8}{\omega_n^2+16}$ and $b_n(x) = A_n(\sin(\omega_n x) + \frac{\omega_n}{4}\cos(\omega_n x))$, ω_n is the n_{th} solution of the equation $\tan(\omega_n) = \frac{8\omega_n}{\omega_n^2-16}$ and A_n is a normalisation constant which makes $\|b_n\| = 1$.

In terms of the inverse problem setting, we are using the same parameters as before ($\boldsymbol{\theta}^\dagger = [0.098, 0.430]$, $d_{\mathbf{y}} = 6$, noise level $\sigma_\eta^2 = 10^{-4}$). The number of training points for all the emulators has been set to $N = 4$ (chosen using the Halton sequence), while in the case of the PDE-constrained emulator we have used $\bar{N} = 10$ and $d_f = 8$. Furthermore, throughout this numerical experiment, we take the prior of the parameters to be the uniform distribution on $[-1, 1]^2$.

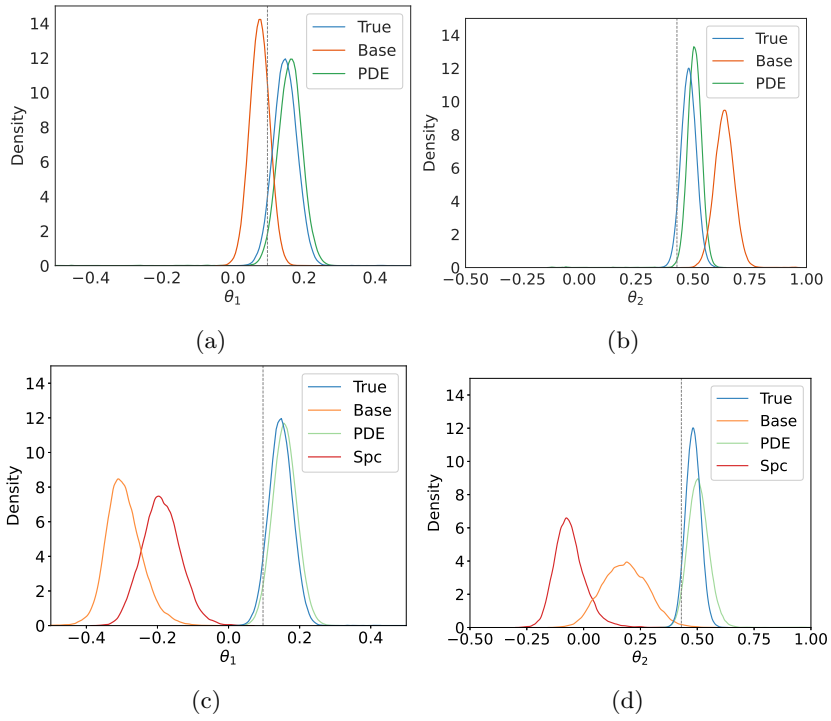


Figure 9: Comparison of different models' marginal distribution when $N = 4$, for PDE model $\bar{N} = 10$ and $d_f = 50$. (a) Mean-based approximation θ_1. (b) Mean-based approximation θ_2. (c) Marginal approximation θ_1. (d) Marginal approximation θ_2. $\mathcal{G}_X$ is the integrals of solution u in (24) with diffusion coefficient (25).

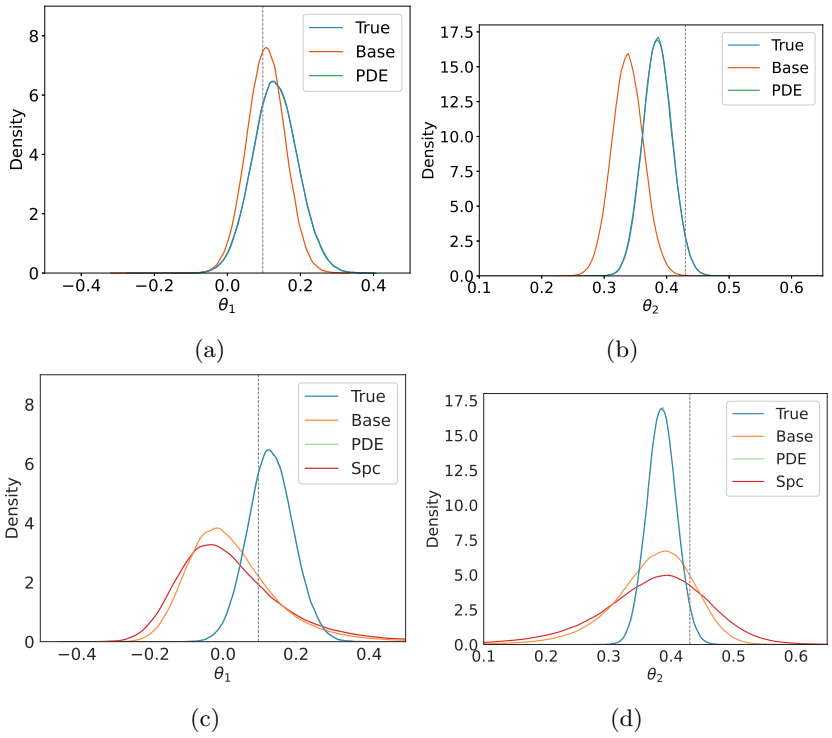


Figure 10: Comparison of different models' marginal distribution when $N = 4$, for PDE model $\bar{N} = 10$ and $d_f = 8$. (a) Mean-based approximation for the θ_1 marginal. (b) Mean-based approximation for the θ_2 marginal. (c) Marginal approximation of the θ_1 marginal. (d) Marginal approximation of the θ_2 marginal. $\mathcal{G}_X$ is the discretised solution u in (24) with diffusion coefficient (26) and $d_{\boldsymbol{\theta}} = 2$.

For the choices of kernels, we use the squared exponential kernel for both k_p and k_s.

As in the previous experiments, we produce 10^6 samples of the posteriors using MALA, and use the results obtained by a mean-based approximation with the baseline model for $N = 10^2$ training points as the ground truth. We now plot in Figure 10 the θ_1 and θ_2 marginals for the different Gaussian emulators both in the case of mean-based and marginal posterior approximations. In particular, as we can see in Figure 10(a)-(b) for the mean-based posterior approximations, the baseline and spatially correlated model fail to capture the true posterior while this is not the case for the PDE-constrained model since the agreement with the true posterior is excellent. When looking at the marginal approximations in Figure 10(c)-(d) we can see that the marginals for the baseline and spatially correlated model move closer towards the true value $\boldsymbol{\theta}^\dagger$ and exhibit variance inflation. This is, however, not the case for the PDE-constrained model since again it is in excellent agreement with the true posterior.

4.1.5 Ten-dimensional parametric expansion diffusion coefficient

We will now increase the dimension of the diffusion coefficient from $\mathbf{d}_\theta = 2$ to $\mathbf{d}_\theta = 10$ in (26), to test the proposed method in a relatively high dimensional space. In particular, we divide the interval $[0, 1]$ into 12 sub-intervals of equal length and fix the value of κ to be 0 and 1 at the 2 ends, respectively. The values of on the remaining ten intervals are our unknown $\boldsymbol{\theta}$. With regard to the inverse problem setting, we set

$$\boldsymbol{\theta}^\dagger = [0.098, 0.430, 0.206, 0.090, -0.153, 0.292, -0.125, 0.784, 0.927, -0.233]$$

and we increase the number of observation points to $d_{\mathbf{y}} = 20$. The level of noise is same as before ($\sigma_\eta^2 = 10^{-4}$). The number of training points for all emulators is again set to be $N = 4$, and for the PDE-constrained emulator we use $\bar{N} = 50$, $d_f = 25$ and $d_g = 2$. For the choices of kernels, we use the squared exponential kernel for both k_p and k_s.

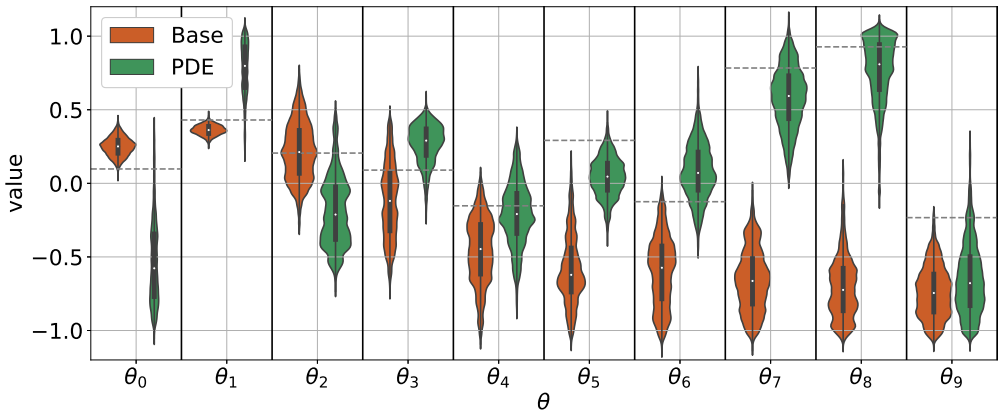
We now use the MALA algorithm to obtain 10^7 samples of the approximate posteriors. In this relatively high-dimensional setting, we need longer chains for the sampling algorithm to converge. Meanwhile, computation of a suitable "ground truth" is prohibitively expensive, so we only compare the sampling result with the true parameter $\boldsymbol{\theta}^\dagger$. The number of training points $N = 4$ is far from enough for the baseline Gaussian process model to give an accurate prediction. From the Figure 11a, we can see that the mean-based posterior approximation with the baseline model can only give a reasonable approximation for the first few variables, for the rest of the variables the approximation could not put any density around the true value. Adding spatial correlation into the model helps the approximation move toward the true value (Figure 11b), but it still cannot correctly approximate the posterior for the last few variables. The performance of the PDE-constrained model is much better than the other models, it is placing the posterior mass around the true value for all variables.

4.2 Two spatial dimensions

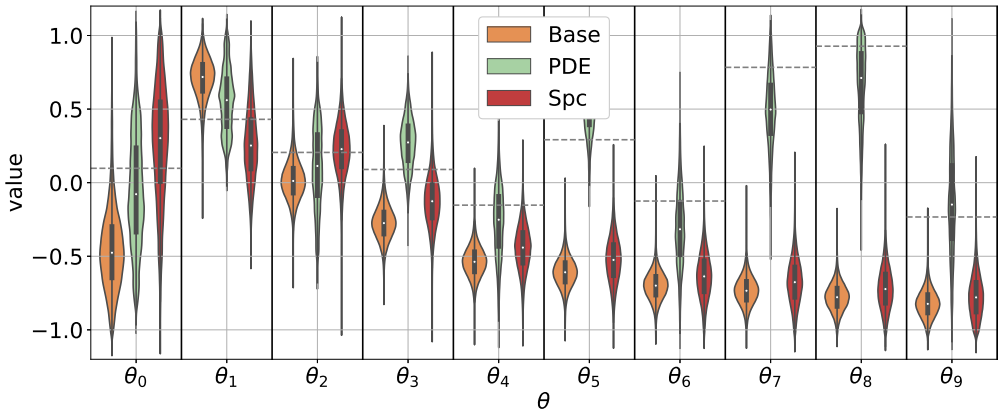
4.2.1 Two-dimensional piece-wise constant diffusion coefficient

In this example, we increase the spatial dimension from $d_{\mathbf{x}} = 1$ to $d_{\mathbf{x}} = 2$ and use a 2 dimensional piece-wise constant as the diffusion coefficient. The values of the diffusion coefficient are set in a similar way to the previous example, depending only on the first dimension of $\mathbf{x}$:

$$\kappa(\mathbf{x}, \boldsymbol{\theta}) = \begin{cases} 0, & \text{for } x_1 \in [0, \frac{1}{4}), \\ \theta_1, & \text{for } x_1 \in [\frac{1}{4}, \frac{1}{2}), \\ \theta_2, & \text{for } x_1 \in [\frac{1}{2}, \frac{3}{4}), \\ 1, & \text{for } x_1 \in [\frac{3}{4}, 1]. \end{cases} \tag{27}$$



(a)



(b)

Figure 11: Comparison of different models' marginal distribution when $N = 4$, for PDE model $\bar{N} = 50$ and $d_f = 25$. (a) Mean-based approximation (b) Marginal approximation. $\mathcal{G}_X$ is the discretised solution u in (24) with diffusion coefficient (26) and $d_{\boldsymbol{\theta}} = 10$.

The boundary conditions are a mixture of Neumann and Dirichlet conditions, given by

$$\begin{aligned} &\partial_{x_1} u(x_1, 0) = \partial_{x_1} u(x_1, 1) = 0, && \text{for } x_1 \in [0, 1], \\ &u(0, x_2) = 1, \quad u(1, x_2) = 0, && \text{for } x_2 \in [0, 1]. \end{aligned}$$

These boundary conditions define a *flow cell*, with no flux at the top and bottom boundary ($x_2 = 0, 1$) and flow from left to right induced by the higher value of u at $x_1 = 0$.

Again, we take the prior of the parameters to be the uniform distribution on $[-1, 1]^2$, approximated by the $\lambda-$Moreau-Yoshida envelope with $\lambda = 10^{-3}$. For the observation, we generate our data $\mathbf{y}$ according to equation (2) for the value $\boldsymbol{\theta}^\dagger = [0.098, 0.430]$ for $d_{\mathbf{y}} = 6$ (chosen to be the first 6 points in the Halton sequence) and for noise level $\sigma_\eta^2 = 10^{-5}$. In addition, for the baseline and spatially correlated model, we have used $N = 4$ training points (chosen to be the first 4 points in the Halton sequence), while additionally for the PDE-constrained model, we have used $\bar{N} = 30$, $d_f = 30$ and $d_g = 8$, corresponding to 2 equally spaced points on each boundary. For the covariance kernels, we let k_p be the squared exponential kernel and k_s be the Matèrn kernel with $\nu = \frac{5}{2}$.

We plot the mean-based approximate posteriors marginals in Figure 12a and 12b. We can see that in this case, the PDE-constrained model significantly improves the approximation accuracy, which is different from the previous piecewise constant diffusion coefficient example in 1 spatial dimension. In Figures 12c and 12d, we compare the marginal approximation for the three models. We see that the PDE-constrained model performs better than the other two models.

4.3 Emulating the negative log-likelihood function

As discussed in Section 2.3.2, we can emulate the negative log-likelihood (also called potential function) directly with Gaussian process regression. Since emulation of log-likelihood simplifies the structure of the problem, we are not able to incorporate spatial correlation or PDE constraints into the emulator. We have mean-based approximation (12) and marginal approximation (13). We test their performance using previous examples: problem (24) with diffusion coefficient (25) with $d_{\mathbf{x}} = 1$ and $d_{\mathbf{x}} = 2$. All parameters are kept the same as in Section 4.1.2 and Section 4.2. Due to its simplified structure, the value of $d_{\mathbf{x}}$ makes no difference for the emulator since the only information taken by the emulator is the training data $\Phi(\Theta)$.

In Figure 13, we compare the mean-based approximation with emulation of the log-likelihood Φ and the observation operator $\mathcal{G}_X$ using baseline model. We see that the results are very different in both examples. For the $d_{\mathbf{x}} = 1$ example, emulating log-likelihood function performs better than emulation of observation with baseline model, the approximated posterior is closer to the true posterior. For the $d_{\mathbf{x}} = 2$ case, its performance is much worse. Hence, emulating the log-likelihood with a small amount of data could be less reliable compared to emulating observation. If we increase the number of training data to $N = 10$

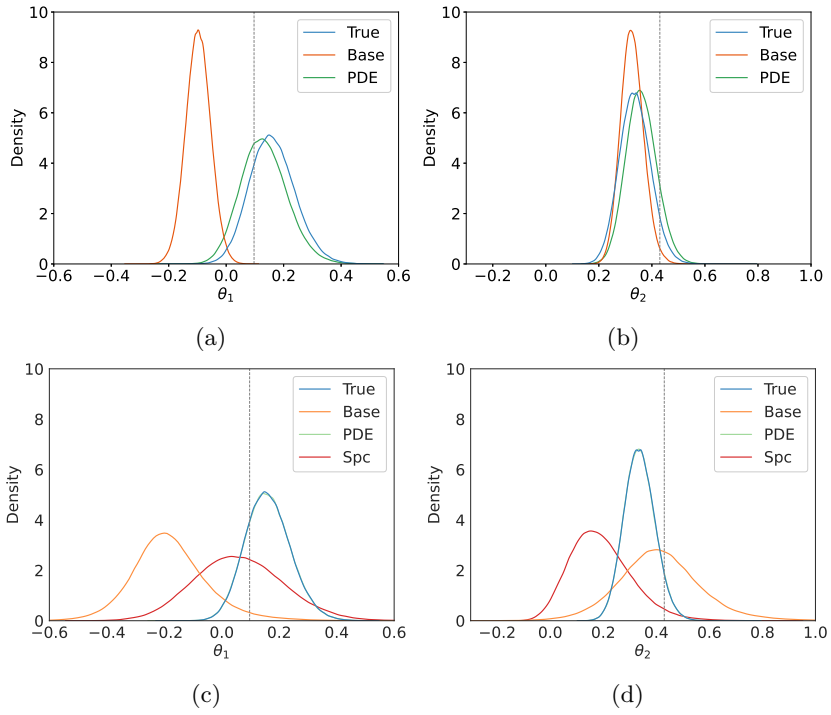
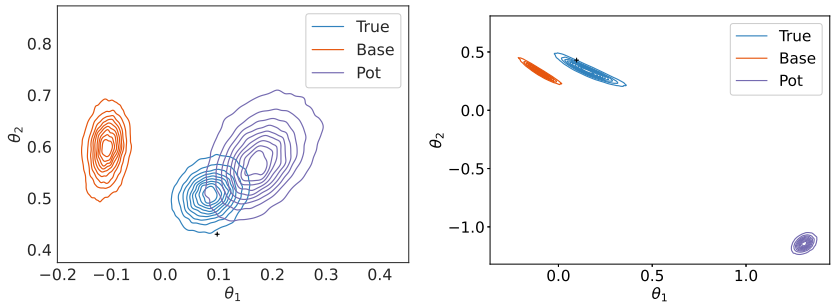


Figure 12: Comparison of different models' marginal distribution when $N = 4$, for PDE model $\bar{N} = 30$ and $d_f = 30$. (a) Mean-based approximation for the θ_1 marginal. (b) Mean-based approximation for the θ_2 marginal. (c) Marginal approximation for the θ_1 marginal. (d) Marginal approximation for the θ_2 marginal. $\mathcal{G}_X$ is the discretised solution u with $d_{\mathbf{x}} = 2$ and diffusion coefficient (27).



(a) (b)

Figure 13: Comparison of emulating log-likelihood function and emulating observations when $N = 4$. Both approximation are mean-based approximation. $\mathcal{G}_X$ is the negative log-likelihood function in: (a) problem (24) with diffusion coefficient (25) with $d_{\mathbf{x}} = 1$; (b) problem (24) with diffusion coefficient (25) with $d_{\mathbf{x}} = 2$.

for the $d_{\mathbf{x}} = 2$ case, we can see the improvement of accuracy (Figure 14), but it is still worse than emulating observation with baseline model.

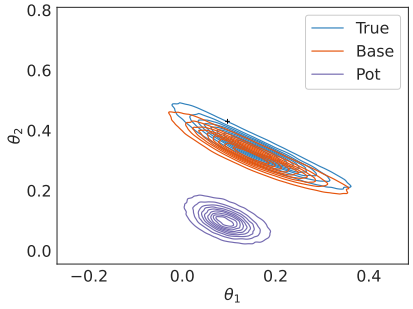
Similarly, marginal approximations of the posterior with emulation of the log-likelihood appear to also be less reliable when the amount of training data is small, see Figure 15. Including more training point can again improve the performance. The main advantage of emulating the log-likelihood function directly is its computational cost, which is much smaller than emulating in observation space. Detailed computational times are listed in the following section.

4.4 Computational timings

In this section, we discuss computational timings. We focus on the computational gains resulting from using Gaussian process emulators instead of the PDE solution in the posterior (see Table 3) and the relative costs of sampling from the various approximate posteriors (see Tables 4, 5 and 6).

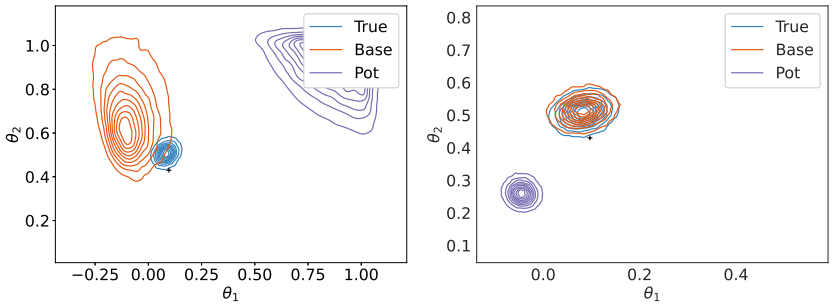
Table 3 below gives average computational timings comparing the evaluation of the solution of the PDE using Firedrake with using the Gaussian process surrogate model. For the baseline surrogate model, the two primary costs are (i) computing the coefficients $\boldsymbol{\alpha} = K(\Theta, \Theta)^{-1}\mathcal{G}_X(\Theta)$, which is an *offline* cost and only needs to be done once, and (ii) computing the predictive mean $m_N^f(\boldsymbol{\theta}) = K(\boldsymbol{\theta}, \Theta)\boldsymbol{\alpha}$, which is the *online* cost and needs to be done for every new test point $\boldsymbol{\theta}$. We see that evaluating $m_N^f(\boldsymbol{\theta})$ is orders of magnitude faster than evaluating $\mathcal{G}_X(\boldsymbol{\theta})$.

In Tables 4, 5 and 6, we compare average computational timings of drawing one sample from the approximate posterior with different models. In Table 4, we see that the mean-based approximation with the PDE-informed prior is more expensive than the one with the baseline prior, by a factor of 2-4 depending on



(a)

Figure 14: Accuracy of emulator is improved when N increases ($N = 10$). $\mathcal{G}_X$ is the negative log-likelihood function in problem (24) with diffusion coefficient (25) with $d_{\mathbf{x}} = 2$ and mean-based approximation.



(a) (b)

Figure 15: (a) Marginal approximation with $N = 4$ (b) Marginal approximation with $N = 10$. $\mathcal{G}_X$ is the negative log-likelihood function in problem (24) with diffusion coefficient (25) with $d_{\mathbf{x}} = 1$.

Set-up	$\mathcal{G}_X(\boldsymbol{\theta})$	$m_N^{\mathcal{G}_X}(\boldsymbol{\theta})$	$\boldsymbol{\alpha}$
$d_{\boldsymbol{\theta}} = 2, d_{\mathbf{y}} = 6, D = (0,1), N = 4$	3.2×10^{-1}s	1.0×10^{-4}s	2.5×10^{-4}s
$d_{\boldsymbol{\theta}} = 2, d_{\mathbf{y}} = 6, D = (0,1), N = 20$	3.2×10^{-1}s	1.3×10^{-4}s	6.8×10^{-4}s
$d_{\boldsymbol{\theta}} = 10, d_{\mathbf{y}} = 18, D = (0,1), N = 4$	3.2×10^{-1}s	1.6×10^{-4}s	4.5×10^{-4}s
$d_{\boldsymbol{\theta}} = 2, d_{\mathbf{y}} = 6, D = (0,1)^2, N = 4$	$7.6 \times 10^{0}s$	$1.0 \times 10^{-4}s$	$5.3 \times 10^{-4}s$

Table 3: Timings of PDE solution vs baseline Gaussian process emulator

the setting. This is to be expected, since the PDE-informed posterior mean $\mathbf{m}_{N,X_f,X_g}^{\mathcal{G}_X}$ involves matrices of larger dimensions than the baseline posterior mean $\mathbf{m}_N^{\mathcal{G}_X}$.

Table 5 investigates the different marginal approximations. Compared to the mean-based approximations in Table 3, we see that the marginal approximations are more expensive by a factor of around 2 for the baseline model and around 3-10 for the PDE-constrained model. Within the different marginal approximations, the spatially correlated model is not much more expensive than the baseline model. Depending on the setting, the PDE-constrained model is 2-30 times more expensive.

In Table 6, we can see that emulating the log-likelihood significantly reduces the cost of sampling from the mean-based and marginal approximations, by around a factor of 20 compared to the baseline model for emulating the observations.

Set-up	$\pi_{\text{mean}}^{N,\mathcal{G}_X}$	$\pi_{\text{mean}}^{N,\mathcal{G}_X,\text{PDE}}$
$d_{\boldsymbol{\theta}} = 2, d_{\mathbf{y}} = 6, D = (0,1), N = 4$	8.5×10^{-4}s	1.2×10^{-3}s ($\bar{N} = 10, d_f = 20$)
$d_{\boldsymbol{\theta}} = 2, d_{\mathbf{y}} = 6, D = (0,1), N = 20$	9.3×10^{-4}s	1.4×10^{-3}s ($\bar{N} = 10, d_f = 20$)
$d_{\boldsymbol{\theta}} = 10, d_{\mathbf{y}} = 18, D = (0,1), N = 4$	2.6×10^{-3}s	1.2×10^{-2}s ($\bar{N} = 50, d_f = 25$)
$d_{\boldsymbol{\theta}} = 2, d_{\mathbf{y}} = 6, D = (0,1)^2, N = 4$	8.5×10^{-4}s	1.6×10^{-3}s ($\bar{N} = 30, d_f = 30$)

Table 4: Timings of different mean-based approximations (baseline and PDE-constrained)

Set-up	$\pi_{\text{marginal}}^{N,\mathcal{G}_X}$	$\pi_{\text{marginal}}^{N,\mathcal{G}_X,s}$	$\pi_{\text{marginal}}^{N,\mathcal{G}_X,\text{PDE}}$
$d_{\boldsymbol{\theta}} = 2, d_{\mathbf{y}} = 6, D = (0,1), N = 4$	1.7×10^{-3}s	2.2×10^{-3}s	3.2×10^{-3}s
$d_{\boldsymbol{\theta}} = 2, d_{\mathbf{y}} = 6, D = (0,1), N = 20$	2.0×10^{-3}s	2.6×10^{-3}s	5.6×10^{-3}s
$d_{\boldsymbol{\theta}} = 10, d_{\mathbf{y}} = 18, D = (0,1), N = 4$	3.4×10^{-3}s	3.6×10^{-3}s	1.1×10^{-1}s
$d_{\boldsymbol{\theta}} = 2, d_{\mathbf{y}} = 6, D = (0,1)^2, N = 4$	1.7×10^{-3}s	2.2×10^{-3}s	4.8×10^{-2}s

Table 5: Timings of different marginal approximations (baseline, spatially correlated and PDE-constrained); $\overline{N}$ and d_f are as in Table 4

Set-up	$\pi^{N,\Phi}_{\text{mean}}$	$\pi^{N,\mathcal{G}_X,\Phi}_{\text{marginal}}$
$d_{\boldsymbol{\theta}} = 2, d_{\mathbf{y}} = 6, D = (0,1), N = 4$	3.4×10^{-5}s	5.8×10^{-5}s
$d_{\boldsymbol{\theta}} = 2, d_{\mathbf{y}} = 6, D = (0,1)^2, N = 4$	3.4×10^{-5}s	5.8×10^{-5}s

Table 6: Timings of mean-based and marginal approximation when emulating the log-likelihood

5 Conclusions, discussion and actionable advice

Bayesian inverse problems in PDEs pose significant computational challenges. Application of state-of-the-art sampling methods, including MCMC methods, is typically computationally infeasible due to the large computational cost of simulating the underlying mathematical model for a given value of the unknown parameters. A solution to alleviate this problem is to use a surrogate model to approximate the PDE solution in Bayesian posterior distribution. In this work we considered the use of Gaussian process surrogate models, which are frequently used in engineering and geo-statistics applications and offer the benefit of built-in uncertainty quantification in the variance of the emulator.

The focus of this work was on practical aspects of using Gaussian process emulators in this context, providing efficient MCMC methods and studying the effect various modelling choices in the derivation of the approximate posterior on its accuracy and computational efficiency. We now summarise the main conclusions of our investigation.

1. **Emulating log-likelihood vs emulating observations.** We can construct an emulator for the negative log-likelihood Φ or the parameter-to-observation map $\mathcal{G}_X$ in the likelihood (3).

   - *Computational efficiency.* The log-likelihood Φ is always scalar-valued, independent of the number of observations $d_{\mathbf{y}}$, which makes the computation of the approximate likelihood for a given value of the parameters $\boldsymbol{\theta}$ much cheaper than the approximate likelihood with emulated $\mathcal{G}_X$. The relative cost will depend on $d_{\mathbf{y}}$.
   - *Accuracy.* When only limited training data is provided, emulating $\mathcal{G}_X$ appears more reliable than emulating Φ, even with the baseline model. The major advantage of emulating $\mathcal{G}_X$ is that it allows us to include correlation between different observations, i.e. between the different entries of $\mathcal{G}_X$. This substantially increases the accuracy of the approximate posteriors, in particular if we use the PDE structure to define the correlations (see point 3 below).

2. **Mean-based vs marginal posterior approximations.** We can use only the mean of the Gaussian process emulator to define the approximate posterior as in (10) and (12), or we can make use of its full distribution to define the marginal approximate posteriors as in (11) and (13).

- *Computational efficiency.* The mean-based approximations are faster to sample from using MALA. This is due to simpler structure of the gradient required for the proposals. The difference in computational time depends on the prior chosen, and is greater for the PDE-constrained model.
- *Accuracy.* The marginal approximations correspond to a form of variance inflation in the approximate posterior (see Section 2.3), representing our incomplete knowledge about the PDE solution. They thus combat over-confident predictions. In our experiments, we confirm that they typically allocate larger mass to regions around the true parameter value than the mean-based approximations.

3. **Spatial correlation and PDE-constrained priors.**

   - *Computational efficiency.* Introducing the spatially correlated model only affects the marginal approximation, and sampling from the marginal approximate posterior with the spatially correlated model is slightly slower than with baseline model. The PDE-constrained model significantly increases the computational times for both the mean-based and marginal approximations, by how much highly depends on the size of additional training data.
   - *Accuracy.* Introducing spatial correlation improves the accuracy of the marginal approximation compared to the baseline model. The most accurate results are obtained with the PDE-constrained priors, which are problem specific and more informative. A benefit of the spatially correlated model is that it does not rely on the underlying PDE being linear, and easily extends to non-linear settings.

In summary, the marginal posterior approximations and spatially correlated/ PDE-constrained prior distributions provide mechanisms of increasing the accuracy of the inference and avoiding over-confident biased predictions, without the need to increase N. This is particularly useful in practical applications, where the number of model runs N available to train the surrogate model may be very small due to constraints in time and/or cost. This does result in higher computational cost compared to mean-based approximations based on black-box priors, but may still be the preferable option if obtaining another training point is impossible or computationally very costly.

Variance inflation, as exhibited in the marginal posterior approximations considered in this work, is a known tool to improve Bayesian inference in complex models, see e.g. [8, 6, 11]. Conceptually, it is also related to including model discrepancy [16, 4]. However, the approach to variance inflation presented in this work has several advantages. Firstly, the variance inflation being equal to the predictive variance of the emulator means that the amount of variance inflation included depends on the location $\boldsymbol{\theta}$ in the parameter space. We introduce more uncertainty in parts of the parameter space where we have less training points and the emulator is possibly less accurate. Secondly, the amount of variance inflation can be tuned in a principled way using standard techniques for

hyper-parameter estimation in Gaussian process emulators. There is no need to choose a model for the variance inflation separately to choosing the emulator, since this is determined automatically as part of the emulator.

We did not discuss optimal experimental design in this work, i.e. how we should optimally choose the locations Θ of the training data. In practice this will also have a large influence on the accuracy of the approximate posteriors, especially for small N. In the context of inverse problems as considered here, one usually wants to place the training points in regions of parameter space where the (approximate) posterior places significant mass (see e.g. [13] and the references therein). For a fair comparison between all scenarios, and to eliminate the interplay between optimal experimental design and other modelling choices, we have chosen the training points as a space-filling design in our experiments. We expect the same conclusions to hold with optimally placed points.

Acknowledgements

The authors would like to thank the Isaac Newton Institute for Mathematical Sciences, Cambridge, for support and hospitality during the programme *Mathematical and statistical foundation of future data-driven engineering* where work on this paper was undertaken. This work was supported by EPSRC grants no EP/R014604/1 and EP/V006177/1.

References

[1] I. Babuska, F. Nobile, and R. Tempone, *A stochastic collocation method for elliptic partial differential equations with random input data*, SIAM J. Numerical Analysis, 45 (2007), pp. 1005–1034.

[2] H. H. Bauschke, R. S. Burachik, P. L. Combettes, V. Elser, D. R. Luke, and H. Wolkowicz, *Fixed-point algorithms for inverse problems in science and engineering*, vol. 49, Springer Science & Business Media, 2011.

[3] S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, *Handbook of Markov Chain Monte Carlo*, CRC press, 2011.

[4] J. Brynjarsdóttir and A. O'Hagan, *Learning about physical parameters: The importance of model discrepancy*, Inverse problems, 30 (2014), p. 114007.

[5] T. Bui-Thanh, K. Willcox, and O. Ghattas, *Model reduction for large-scale systems with high-dimensional parametric input space*, SIAM Journal on Scientific Computing, 30 (2008), pp. 3270–3288.

[6] D. Calvetti, M. Dunlop, E. Somersalo, and A. Stuart, *Iterative updating of model error for bayesian inversion*, Inverse Problems, 34 (2018), p. 025008.

[7] J. Cockayne, C. Oates, T. Sullivan, and M. Girolami, *Probabilistic numerical methods for PDE-constrained Bayesian inverse problems*, AIP Conference Proceedings, 1853 (2017), p. 060001.

[8] P. R. Conrad, M. Girolami, S. Särkkä, A. Stuart, and K. Zygalakis, *Statistical analysis of differential equations: introducing probability measures on numerical solutions*, Statistics and Computing, 27 (2017), pp. 1065–1082.

[9] P. G. Constantine, *Active Subspaces*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2015.

[10] P. G. Constantine, E. Dow, and Q. Wang, *Active subspace methods in theory and practice: Applications to kriging surfaces*, SIAM Journal on Scientific Computing, 36 (2014), pp. A1500–A1524.

[11] C. Fox, T. Cui, and M. Neumayer, *Randomized reduced forward models for efficient metropolis–hastings mcmc, with application to subsurface fluid flow and capacitance tomography*, GEM-International Journal on Geomathematics, 11 (2020), pp. 1–38.

[12] M. Giordano and R. Nickl, *Consistency of bayesian inference with gaussian process priors in an elliptic inverse problem*, Inverse Problems, 36 (2020), p. 085001.

[13] T. Helin, A. M. Stuart, A. L. Teckentrup, and K. C. Zygalakis, *Introduction To Gaussian Process Regression In Bayesian Inverse Problems, With New Results On Experimental Design For Weighted Error Measures*, arXiv preprint arXiv:2302.04518, (2023).

[14] D. Higdon, M. Kennedy, J. C. Cavendish, J. A. Cafeo, and R. D. Ryne, *Combining field data and computer simulations for calibration and prediction*, SIAM Journal on Scientific Computing, 26 (2004), pp. 448–466.

[15] J. Kaipio and E. Somersalo, *Statistical and Computational Inverse Problems*, Springer, Dordrecht, 2005.

[16] M. C. Kennedy and A. O'Hagan, *Bayesian calibration of computer models*, Journal of the Royal Statistical Society, Series B, Methodological, 63 (2000), pp. 425–464.

[17] H. C. Lie, T. J. Sullivan, and A. L. Teckentrup, *Random forward models and log-likelihoods in Bayesian inverse problems*, SIAM/ASA Journal on Uncertainty Quantification, 6 (2018), pp. 1600–1629.

[18] Y. Marzouk and D. Xiu, *A stochastic collocation approach to bayesian inference in inverse problems*, PRISM: NNSA Center for Prediction of Reliability, Integrity and Survivability of Microsystems, 6 (2009).

[19] Y. M. Marzouk, H. N. Najm, and L. A. Rahn, *Stochastic spectral methods for efficient bayesian solution of inverse problems*, Journal of Computational Physics, 224 (2007), pp. 560–586.

[20] T. Matsumoto and T. Sullivan, *Images of Gaussian and other stochastic processes under closed, densely-defined, unbounded linear operators*, arXiv preprint arXiv:2305.03594, (2023).

[21] H. Niederreiter, *Random number generation and quasi-Monte Carlo methods*, SIAM, 1992.

[22] A. O'Hagan, *Bayesian analysis of computer code outputs: A tutorial*, Reliability Engineering & System Safety, 91 (2006), pp. 1290–1300.

[23] M. Raissi, P. Perdikaris, and G. E. Karniadakis, *Machine learning of linear differential equations using gaussian processes*, Journal of Computational Physics, 348 (2017), pp. 683–693.

[24] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, MIT Press, 2006.

[25] F. Rathgeber, D. A. Ham, L. Mitchell, M. Lange, F. Luporini, A. T. McRae, G.-T. Bercea, G. R. Markall, and P. H. Kelly, *Firedrake: automating the finite element method by composing abstractions*, ACM Transactions on Mathematical Software (TOMS), 43 (2016), pp. 1–27.

[26] C. Robert and G. Casella, *Monte Carlo statistical methods*, Springer Verlag, 2004.

[27] G. O. Roberts and R. L. Tweedie, *Exponential convergence of langevin distributions and their discrete approximations*, Bernoulli, (1996), pp. 341–363.

[28] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, *Design and Analysis of Computer Experiments*, Statistical Science, 4 (1989), pp. 409 – 423.

[29] M. Spitieris and I. Steinsland, *Bayesian Calibration of Imperfect Computer Models using Physics-Informed Priors*, Journal of Machine Learning Research, 24 (2023), pp. 1–39.

[30] M. L. Stein, *Interpolation of spatial data*, Springer Series in Statistics, Springer-Verlag, New York, 1999.

[31] A. Stuart and A. Teckentrup, *Posterior consistency for gaussian process approximations of bayesian posterior distributions*, Mathematics of Computation, 87 (2018).

[32] A. M. Stuart, *Inverse problems: A bayesian perspective*, Acta Numerica, 19 (2010), p. 451–559.

[33] L. P. Swiler, M. Gulian, A. L. Frankel, C. Safta, and J. D. Jakeman, *Constrained gaussian processes: A survey.*, tech. rep., Sandia National Lab.(SNL-NM), Albuquerque, NM (United States); Sandia ..., 2021.

[34] A. L. Teckentrup, *Convergence of gaussian process regression with estimated hyper-parameters and applications in bayesian inverse problems*, SIAM/ASA Journal on Uncertainty Quantification, 8 (2020), pp. 1310–1337.

[35] D. Xiu and G. E. Karniadakis, *Modeling uncertainty in flow simulations via generalized polynomial chaos*, Journal of Computational Physics, 187 (2003), pp. 137–167.

A Derivation of the analytical formula of the marginal approximation

In order to simplify the notation here, we let $\mathbf{m}_{\boldsymbol{\theta}} = \mathbf{m}_N^{\mathcal{G}_X}(\boldsymbol{\theta})$, $K_{\boldsymbol{\theta}} = K_N(\boldsymbol{\theta}, \boldsymbol{\theta})$ and $\Gamma_\eta = \sigma_\eta^{-2} I_{d_{\mathbf{y}}}$. First, we assume $\mathcal{G}_X^N(\boldsymbol{\theta}) = \mathbf{m}_{\boldsymbol{\theta}} + \boldsymbol{\xi}$, where $\boldsymbol{\xi} \sim \mathcal{N}(0, K_{\boldsymbol{\theta}})$, so by the definition of expectation we have

$$\begin{aligned}
&\mathbb{E}\left(\exp\left(-\frac{1}{2}\|\mathcal{G}_X^N(\boldsymbol{\theta}) - \mathbf{y}\|_{\Gamma_e ta}\right)\pi_0(\boldsymbol{\theta})\right) \\
&= \frac{1}{\sqrt{(2\pi)^{d_{\mathbf{y}}}\det(K_{\boldsymbol{\theta}})}} \int_{\mathbb{R}^{d_{\mathbf{y}}}} \exp\left(-\frac{\|\mathbf{m}_{\boldsymbol{\theta}} + \boldsymbol{\xi} - \mathbf{y}\|_{\Gamma_\eta}^2}{2}\right) \exp\left(-\frac{\|\boldsymbol{\xi}\|_{K_{\boldsymbol{\theta}}}^2}{2}\right) d\boldsymbol{\xi},
\end{aligned}$$

then rewrite and simplify the formula

$$= \frac{1}{\sqrt{(2\pi)^{d_{\mathbf{y}}}\det(K_{\boldsymbol{\theta}})}} \int_{\mathbb{R}^{d_{\mathbf{y}}}} \exp\left(-\frac{1}{2}\left(\|\boldsymbol{\xi} - (\mathbf{y} - \mathbf{m}_{\boldsymbol{\theta}})\|_{\Gamma_\eta}^2 + \|\boldsymbol{\xi}\|_{K_{\boldsymbol{\theta}}}^2\right)\right) d\boldsymbol{\xi}$$

we let $\bar{\mathbf{y}} = \mathbf{y} - \mathbf{m}_{\boldsymbol{\theta}}$, then

$$\begin{aligned}
&= \frac{1}{\sqrt{(2\pi)^{d_{\mathbf{y}}}\det(K_{\boldsymbol{\theta}})}} \int_{\mathbb{R}^{d_{\mathbf{y}}}} \exp\left(-\frac{1}{2}\left(\|\boldsymbol{\xi} - \bar{\mathbf{y}}\|_{\Gamma_\eta}^2 + \|\boldsymbol{\xi}\|_{K_{\boldsymbol{\theta}}}^2\right)\right) d\boldsymbol{\xi} \\
&= \frac{1}{\sqrt{(2\pi)^{d_{\mathbf{y}}}\det(K_{\boldsymbol{\theta}})}} \int_{\mathbb{R}^{d_{\mathbf{y}}}} \exp\left(-\frac{1}{2}\left((\boldsymbol{\xi} - \bar{\mathbf{y}})^T\Gamma_\eta^{-1}(\boldsymbol{\xi} - \bar{\mathbf{y}}) + \boldsymbol{\xi}^T K_{\boldsymbol{\theta}}^{-1}\boldsymbol{\xi}\right)\right) d\boldsymbol{\xi} \\
&= \frac{1}{\sqrt{(2\pi)^{d_{\mathbf{y}}}\det(K_{\boldsymbol{\theta}})}} \int_{\mathbb{R}^{d_{\mathbf{y}}}} \exp\left(-\frac{1}{2}\left(\boldsymbol{\xi}^T(\Gamma_\eta^{-1} + K_{\boldsymbol{\theta}}^{-1})\boldsymbol{\xi} - 2\bar{\mathbf{y}}^T\Gamma_\eta^{-1}\boldsymbol{\xi} + \bar{\mathbf{y}}^T\Gamma_\eta^{-1}\bar{\mathbf{y}}\right)\right) d\boldsymbol{\xi}
\end{aligned}$$

Since Γ_η and $K_{\boldsymbol{\theta}}$ are symmetric matrices, we have

$$\begin{aligned}
\bar{\mathbf{y}}^T\Gamma_\eta^{-1}\boldsymbol{\xi} &= \bar{\mathbf{y}}^T((K_{\boldsymbol{\theta}} + \Gamma_\eta)^{-1}K_{\boldsymbol{\theta}})(K_{\boldsymbol{\theta}}^{-1}(K_{\boldsymbol{\theta}} + \Gamma_\eta))\Gamma_\eta^{-1}\boldsymbol{\xi} \\
&= (K_{\boldsymbol{\theta}}(K_{\boldsymbol{\theta}} + \Gamma_\eta)^{-1}\bar{\mathbf{y}})^T K_{\boldsymbol{\theta}}^{-1}(K_{\boldsymbol{\theta}} + \Gamma_\eta)\Gamma_\eta^{-1}\boldsymbol{\xi} \\
&= \tilde{\mathbf{y}}^T C^{-1}\boldsymbol{\xi},
\end{aligned}$$

where $C = K_{\boldsymbol{\theta}}(K_{\boldsymbol{\theta}} + \Gamma_\eta)^{-1}\Gamma_\eta$ and $\tilde{\mathbf{y}} = C\Gamma_\eta^{-1}\bar{\mathbf{y}}$. Substituting it into the formula above, we have

$$= \frac{1}{\sqrt{(2\pi)^{d_{\mathbf{y}}}\det(K_{\boldsymbol{\theta}})}} \int_{\mathbb{R}^{d_{\mathbf{y}}}} \exp\left(-\frac{1}{2}\left(\boldsymbol{\xi}^T C^{-1}\boldsymbol{\xi} - 2\tilde{\mathbf{y}}^T C^{-1}\boldsymbol{\xi} + \bar{\mathbf{y}}^T\Gamma_\eta^{-1}\bar{\mathbf{y}}\right)\right) d\boldsymbol{\xi}$$

Then we can complete the square

$$
\begin{aligned}
&= \frac{1}{\sqrt{(2\pi)^{d_{\mathbf{y}}} \det\left(K_{\boldsymbol{\theta}}\right)}} \int_{\mathbb{R}^{d_{\mathbf{y}}}} \exp\left(-\frac{1}{2}\left(\|\boldsymbol{\xi}-\tilde{\mathbf{y}}\|_C^2 - \tilde{\mathbf{y}}^T C^{-1}\tilde{\mathbf{y}} + \bar{\mathbf{y}}^T \Gamma_\eta^{-1}\bar{\mathbf{y}}\right)\right) d\boldsymbol{\xi} \\
&= \frac{1}{\sqrt{(2\pi)^{d_{\mathbf{y}}} \det\left(K_{\boldsymbol{\theta}}\right)}} \int_{\mathbb{R}^{d_{\mathbf{y}}}} \exp\left(-\frac{1}{2}\left(\|\boldsymbol{\xi}-\tilde{\mathbf{y}}\|_C^2 - (C\Gamma_\eta^{-1}\bar{\mathbf{y}})^T C^{-1}(C\Gamma_\eta^{-1}\bar{\mathbf{y}}) + \bar{\mathbf{y}}^T \Gamma_\eta^{-1}\bar{\mathbf{y}}\right)\right) d\boldsymbol{\xi} \\
&= \frac{1}{\sqrt{(2\pi)^{d_{\mathbf{y}}} \det\left(K_{\boldsymbol{\theta}}\right)}} \int_{\mathbb{R}^{d_{\mathbf{y}}}} \exp\left(-\frac{1}{2}\left(\|\boldsymbol{\xi}-\tilde{\mathbf{y}}\|_C^2 - \bar{\mathbf{y}}^T \Gamma_\eta^{-1} K_{\boldsymbol{\theta}}(K_{\boldsymbol{\theta}}+\Gamma_\eta)^{-1}\bar{\mathbf{y}}) + \bar{\mathbf{y}}^T \Gamma_\eta^{-1}\bar{\mathbf{y}}\right)\right) d\boldsymbol{\xi} \\
&= \frac{1}{\sqrt{(2\pi)^{d_{\mathbf{y}}} \det\left(K_{\boldsymbol{\theta}}\right)}} \int_{\mathbb{R}^{d_{\mathbf{y}}}} \exp\left(-\frac{1}{2}\left(\|\boldsymbol{\xi}-\tilde{\mathbf{y}}\|_C^2 - \bar{\mathbf{y}}^T (\Gamma_\eta^{-1} K_{\boldsymbol{\theta}}(K_{\boldsymbol{\theta}}+\Gamma_\eta)^{-1} - \Gamma_\eta^{-1})\bar{\mathbf{y}})\right)\right) d\boldsymbol{\xi} \\
&= \frac{1}{\sqrt{(2\pi)^{d_{\mathbf{y}}} \det\left(K_{\boldsymbol{\theta}}\right)}} \int_{\mathbb{R}^{d_{\mathbf{y}}}} \exp\left(-\frac{1}{2}\left(\|\boldsymbol{\xi}-\tilde{\mathbf{y}}\|_C^2 + \bar{\mathbf{y}}^T (K_{\boldsymbol{\theta}}+\Gamma_\eta)^{-1}\bar{\mathbf{y}}\right)\right) d\boldsymbol{\xi} \\
&= \frac{1}{\sqrt{(2\pi)^{d_{\mathbf{y}}} \det\left(K_{\boldsymbol{\theta}}\right)}} \exp\left(-\frac{1}{2}\|\bar{\mathbf{y}}\|^2_{(K_{\boldsymbol{\theta}}+\Gamma_\eta)}\right) \int_{\mathbb{R}^{d_{\mathbf{y}}}} \exp\left(-\frac{1}{2}\left(\|\boldsymbol{\xi}-\tilde{\mathbf{y}}\|_C^2\right)\right) d\boldsymbol{\xi} \\
&= \frac{\sqrt{\det\left(C\right)}}{\sqrt{\det\left(K_{\boldsymbol{\theta}}\right)}} \exp\left(-\frac{1}{2}\|\bar{\mathbf{y}}\|^2_{(K_{\boldsymbol{\theta}}+\Gamma_\eta)}\right) \int_{\mathbb{R}^{d_{\mathbf{y}}}} \frac{1}{\sqrt{(2\pi)^{d_{\mathbf{y}}} \det\left(C\right)}} \exp\left(-\frac{1}{2}\left(\|\boldsymbol{\xi}-\tilde{\mathbf{y}}\|_C^2\right)\right) d\boldsymbol{\xi} \\
&\propto \frac{1}{\sqrt{(2\pi)^{d_{\mathbf{y}}} \det(K_{\boldsymbol{\theta}}+\Gamma_\eta)}} \exp\left(-\frac{1}{2}\|\mathbf{y}-\mathbf{m}_{\boldsymbol{\theta}}\|^2_{(K_{\boldsymbol{\theta}}+\Gamma_\eta)}\right)
\end{aligned}
$$

Hence, we obtain the explicit form of the marginal approximation.

B Derivation of the gradient of the approximate log-posteriors

Proof.

$$
\begin{aligned}
\nabla \log \pi^{N,\mathcal{G}_X}_{\text{mean}}(\boldsymbol{\theta}|\mathbf{y}) &= \nabla \log\left(\exp\left(-\frac{1}{2\sigma_\eta^2}\|\mathbf{m}_N^{\mathcal{G}_X}(\boldsymbol{\theta}) - \mathbf{y}\|^2\right)\right) \\
&= -\frac{1}{2\sigma_\eta^2}\nabla\left(\|\mathbf{m}_N^{\mathcal{G}_X}(\boldsymbol{\theta}) - \mathbf{y}\|^2\right) \\
&= -\frac{1}{\sigma_\eta^2}\left(\nabla \mathbf{m}_N^{\mathcal{G}_X}(\boldsymbol{\theta})\right)^T\left(\mathbf{m}_N^{\mathcal{G}_X}(\boldsymbol{\theta}) - \mathbf{y}\right) \\
&= -\frac{1}{\sigma_\eta^2}\left(\nabla K(\boldsymbol{\theta},\Theta)K(\Theta,\Theta)^{-1}\mathbf{y}\right)^T\left(\mathbf{m}_N^{\mathcal{G}_X}(\boldsymbol{\theta}) - \mathbf{y}\right)
\end{aligned}
$$

$$
\begin{aligned}
&\nabla \log \pi^{N,\mathcal{G}_X}_{\text{marginal}}(\boldsymbol{\theta}|\mathbf{y}) \\
&= \nabla \log \left(\frac{\exp\left(-\frac{1}{2}\|\mathbf{m}_N^{\mathcal{G}_X}(\boldsymbol{\theta}) - \mathbf{y}\|^2_{(K_N(\boldsymbol{\theta},\boldsymbol{\theta})+\Gamma_\eta)}\right)}{\sqrt{(2\pi)^{d_{\mathbf{y}}} \det\left(K_N(\boldsymbol{\theta},\boldsymbol{\theta}) + \Gamma_\eta\right)}} \right) \\
&= -\frac{1}{2}\nabla\left(\|\mathbf{m}_N^{\mathcal{G}_X}(\boldsymbol{\theta}) - \mathbf{y}\|^2_{(K_N(\boldsymbol{\theta},\boldsymbol{\theta})+\Gamma_\eta)}\right) - \frac{1}{2}\nabla \log\left((2\pi)^n \det\left(K_N(\boldsymbol{\theta},\boldsymbol{\theta}) + \Gamma_\eta\right)\right) \\
&= -(\nabla K(\boldsymbol{\theta},\Theta)K(\Theta,\Theta)^{-1}\mathbf{y})^T (K_N(\boldsymbol{\theta},\boldsymbol{\theta}) + \Gamma_\eta)^{-1}(\mathbf{m}_N^{\mathcal{G}_X}(\boldsymbol{\theta}) - \mathbf{y}) \\
&\quad - \frac{1}{2}(\mathbf{m}_N^{\mathcal{G}_X}(\boldsymbol{\theta}) - \mathbf{y})^T \nabla\left((K_N(\boldsymbol{\theta},\boldsymbol{\theta}) + \Gamma_\eta)^{-1}\right)(\mathbf{m}_N^{\mathcal{G}_X}(\boldsymbol{\theta}) - \mathbf{y}) \\
&\quad - \frac{1}{2}\left(\operatorname{Tr}\left((K_N(\boldsymbol{\theta},\boldsymbol{\theta}) + \Gamma_\eta)^{-1}\right)\nabla(K_N(\boldsymbol{\theta},\boldsymbol{\theta}))\right),
\end{aligned}
$$

where

$$
\nabla\left((K_N(\boldsymbol{\theta},\boldsymbol{\theta}) + \Gamma_\eta)^{-1}\right) = -(K_N(\boldsymbol{\theta},\boldsymbol{\theta}) + \Gamma_\eta)^{-1}\nabla\left(K_N(\boldsymbol{\theta},\boldsymbol{\theta})\right)(K_N(\boldsymbol{\theta},\boldsymbol{\theta}) + \Gamma_\eta)^{-1},
$$

and

$$
\nabla K_N(\boldsymbol{\theta},\boldsymbol{\theta}) = 2\nabla K(\boldsymbol{\theta},\Theta)K(\Theta,\Theta)^{-1}K(\Theta,\boldsymbol{\theta})
$$

□