Statistics and Computing (2024) 34:139
https://doi.org/10.1007/s11222-024-10452-2

ORIGINAL PAPER

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

Tianming Bai[1] · **Aretha L. Teckentrup**[1] · **Konstantinos C. Zygalakis**[1]

Received: 19 March 2024 / Accepted: 10 June 2024 / Published online: 24 June 2024

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.

Keywords Bayesian inverse problem · Gaussian process regression · MCMC · Surrogate model

1 Introduction

Combining complex mathematical models with observational data is an extremely challenging yet ubiquitous problem in the fields 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 observations of model outputs, are hence of frequent interest. Adopting a Bayesian approach (Kaipio and Somersalo 2005; Stuart 2010), 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 chain Monte Carlo (MCMC) (Robert and Casella 2004; Brooks et al. 2011) 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. 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 (Rasmussen and Williams 2006; Stein 1999; Sacks et al. 1989; Kennedy and O'Hagan 2000; O'Hagan 2006; Higdon et al. 2004), but other types of surrogate models can also be used including projection-based methods (Bui-Thanh et al. 2008), generalised Polynomial Chaos (Xiu and Karniadakis 2003; Marzouk et al. 2007), sparse grid collocation (Babuska et al. 2007; Marzouk and Xiu 2009) and adaptive subspace methods (Constantine 2015; Constantine et al. 2014).

✉ Tianming Bai
tianming.bai@ed.ac.uk

Aretha L. Teckentrup
a.teckentrup@ed.ac.uk

Konstantinos C. Zygalakis
k.zygalakis@ed.ac.uk

[1] School of Mathematics and Maxwell Institute for Mathematical Sciences, University of Edinburgh, Edinburgh, UK

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 is related to the solution of a linear partial differential equation (PDE). In particular, we consider two different ways of using the surrogate model, 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 Stuart and Teckentrup (2018); Teckentrup (2020); Helin et al. (2023). 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 main setting of 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 Gaussian prior as informative as possible, by incorporating known information about the underlying forward model. For example, such a Gaussian prior specially tailored to solving the forward problem in linear PDEs can be found in Raissi et al. (2017). For incorporating more general constraints, we refer the reader to the recent review (Swiler et al. 2021). 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. Stuart and Teckentrup (2018); Lie et al. (2018)).

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 Raissi et al. (2017) 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 of 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 Sect. 2 we set up notation with respect to the inverse problems of interest and discuss the different kinds of posterior approximations that result from using Gaussian surrogate models for the data-likelihood. We then proceed in Sect. 3 to present our main methodology, discussing how one can 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 Sect. 4, and finally Sect. 5 provides a summary and discussion of the main results.

2 Preliminaries

We now give more details about the type of inverse problems considered in this paper and discuss different aspects of Gaussian emulators, as well as the corresponding type of approximate posteriors considered 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 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 $\mathcal{G}_X(\boldsymbol{\theta}) =$

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

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
$\boldsymbol{\eta}, \Gamma_\eta, \sigma_\eta^2$	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	Hyperparameters in Gaussian covariance function
$k(\boldsymbol{\theta}, \boldsymbol{\theta}')$	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
$\pi_{\text{mean}}^{N,\mathcal{G}_X}, \pi_{\text{marginal}}^{N,\mathcal{G}_X}$	Mean-based and marginal posterior with baseline
$\pi_{\text{mean}}^{N,\mathcal{G}_X,s}, \pi_{\text{marginal}}^{N,\mathcal{G}_X,s}$	Mean-based and marginal posterior with spatial correlation
$\pi_{\text{mean}}^{N,\mathcal{G}_X,\text{PDE}}, \pi_{\text{marginal}}^{N,\mathcal{G}_X,\text{PDE}}$	Mean-based and marginal posterior with 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

$\{u(\mathbf{x}_j; \boldsymbol{\theta})\}_{j=1}^{d_{\mathbf{y}}}$, 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 (Lie et al. 2018). Likewise, the methodology generalises straightforwardly to general bounded linear observation operators applied to the PDE solution u (see the discussion in Sect. 3.1).

To solve the inverse problem we will adopt a Bayesian approach (Stuart 2010). 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

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

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}. Note that our notation of $\|\mathbf{z}\|_{\Gamma_\eta}$ here follows the convention introduced in Stuart (2010). Then by Bayes' formula we have

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

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 nonparametric model for Bayesian inference (Rasmussen and Williams 2006). Our starting point for approximating an arbi-

trary function $\mathbf{g} : \mathcal{T} \to \mathbb{R}^d$, for some $d \in \mathbb{N}$ is the Gaussian process prior

$$\mathbf{g}_0(\boldsymbol{\theta}) \sim \mathrm{GP}(\mathbf{m}(\boldsymbol{\theta}), K(\boldsymbol{\theta}, \boldsymbol{\theta}')), \tag{4}$$

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 positive definite covariance function which represents the covariance between the different entries of $\mathbf{g}$ evaluated at $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$. Distinct from the prior introduced earlier in solving the Bayesian inverse problem, this prior is built for Gaussian process regression. 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, $d = 1$ when directly emulating the log-likelihood.

In the case where $d > 1$ there is a number of choices that one can make for the matrix-valued covariance function (Alvarez et al. 2012). In this section, for simplicity we will assume that the matrix $K(\boldsymbol{\theta}, \boldsymbol{\theta}')$ takes the form

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

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. We will refer to this as the baseline model. 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 (Rasmussen and Williams 2006)

$$k(\boldsymbol{\theta}, \boldsymbol{\theta}') = \sigma^2 \exp\left(-\frac{\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|^2}{2l^2}\right), \tag{5}$$

and the Matérn covariance function (Rasmussen and Williams 2006)

$$k(\boldsymbol{\theta}, \boldsymbol{\theta}') = \frac{\sigma^2}{\Gamma(\nu)2^{\nu-1}}\left(\sqrt{2\nu}\frac{\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|}{l}\right)^{\nu} B_{\nu}\left(\sqrt{2\nu}\frac{\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|}{l}\right). \tag{6}$$

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 hyperparameter $\nu > 0$. 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 $\Theta = \{\boldsymbol{\theta}^i\}_{i=1}^N \subseteq \mathbb{R}^{d_{\boldsymbol{\theta}}}$ with corresponding function values

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

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

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

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 (Rasmussen and Williams 2006). In particular, if we denote with $\mathbf{g}^N$ the Gaussian process (4) 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

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

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

with

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

and

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

We note that $\mathbf{g}^N$ is the Gaussian process posterior, but to avoid confusion with the posterior of the Bayesian inverse problem, we call it the predictive Gaussian process. In addition, for clarity of 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 $\mathcal{G}_X(\Theta) = \{\mathcal{G}_X(\boldsymbol{\theta}^i)\}_{i=1}^N$, we can now proceed with building our Gaussian process emulator for the forward map $\mathcal{G}_X$. Therefore, using similar notation to (7), we denote the predictive Gaussian process by $\mathcal{G}_X^N$. 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

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

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]:

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

where the expectation is taken over the probability space of the Gaussian process posterior. A detailed derivation of (11) can be found in Appendix A. 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 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.

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 evaluations $\boldsymbol{\Phi}(\Theta) = \{\Phi(\boldsymbol{\theta}^i,\mathbf{y})\}_{i=1}^N$, and obtain the predictive Gaussian process $\Phi^N(\boldsymbol{\theta}) \sim \text{GP}(m_N^\Phi(\boldsymbol{\theta}), k_N(\boldsymbol{\theta},\boldsymbol{\theta}))$. 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 the true posterior: 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 approximation in total. The convergence properties of all these approximate posteriors were the subject of study in Stuart and Teckentrup (2018); Teckentrup (2020); Helin et al. (2023), 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 locations 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. Note that when emulating the potential Φ instead of the forward map $\mathcal{G}_X$, we are emulating a

[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 A for completeness, the formula was also derived in Cockayne et al. (2017); Calvetti et al. (2018).

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.

Introducing spatial correlation when emulating $\mathcal{G}_X(\boldsymbol{\theta})$ 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 know that our forward map is associated with the solution of a linear PDE. We do this in Sect. 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 Sect. 3.2.

3.1 Correlated and PDE-informed priors

We now discuss two different approaches to incorporate spatial correlation into 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 $\mathbb{R}^{d_Y}$, 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.

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 obtain 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 conditioned on training data $\mathcal{G}_X(\Theta)$, and due to the separable structure in (14), the predictive mean $\mathbf{m}_N^{\mathcal{G}_X}(\boldsymbol{\theta})$ is in fact the same as for the baseline model in Sect. 2.2. See Appendix B for details.

The second way of introducing spatial correlation is explicitly taking 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 described in Sect. 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 $\mathbf{x}_g \in \partial D$ we have that

$$\begin{bmatrix} u(\boldsymbol{\theta}, \mathbf{x}) \\ g(\boldsymbol{\theta}, \mathbf{x}_g) \\ f(\boldsymbol{\theta}, \mathbf{x}_f) \end{bmatrix} \sim \mathrm{GP}\left(\mathbf{0}, k_p(\boldsymbol{\theta}, \boldsymbol{\theta}') \begin{bmatrix} k_s(\mathbf{x}, \mathbf{x}) & \mathcal{B}k_s(\mathbf{x}, \mathbf{x}_g) & \mathcal{L}^{\boldsymbol{\theta}'}k_s(\mathbf{x}, \mathbf{x}_f) \\ \mathcal{B}k_s(\mathbf{x}_g, \mathbf{x}) & \mathcal{B}\mathcal{B}k_s(\mathbf{x}_g, \mathbf{x}_g) & \mathcal{B}\mathcal{L}^{\boldsymbol{\theta}'}k_s(\mathbf{x}_g, \mathbf{x}_f) \\ \mathcal{L}^{\boldsymbol{\theta}}k_s(\mathbf{x}_f, \mathbf{x}) & \mathcal{L}^{\boldsymbol{\theta}}\mathcal{B}k_s(\mathbf{x}_f, \mathbf{x}_g) & \mathcal{L}^{\boldsymbol{\theta}}\mathcal{L}^{\boldsymbol{\theta}'}k_s(\mathbf{x}_f, \mathbf{x}_f) \end{bmatrix}\right), \tag{16}$$

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. (Matsumoto and Sullivan 2023)) in the derivation. The idea of a joint prior between u and f was also used in Raissi et al. (2017); Spitieris and Steinsland (2023); Cockayne et al. (2017); Pförtner et al. (2022), while (Chris J. Oates and Girolami 2019) uses this explicitly in an inverse problem setting. The crucial difference is that in these works u and f were considered as functions of the spatial variable $\mathbf{x}$ only, while here we instead explicitly model the dependency of u on $\boldsymbol{\theta}$. We then have

$$\begin{bmatrix} \mathcal{G}_X(\boldsymbol{\theta}) \\ g(\boldsymbol{\theta}, X_g) \\ f(\boldsymbol{\theta}, X_f) \end{bmatrix} \sim \mathrm{GP}\left(\mathbf{0}, K(\boldsymbol{\theta}, \boldsymbol{\theta}')\right), \tag{17}$$

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).

The prior (17) can then again be conditioned on training data as in Sect. 2.2, see Appendix B for details. Note that in this case 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 $\mathcal{G}_X(\Theta)$, essentially augmenting the space on which the emulator $\mathcal{G}_X^N(\boldsymbol{\theta})$ is trained. Since g and f are assumed known, these additional observations are cheap to obtain. It is also possible 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 Sect. 4 and Appendix D).

3.2 MCMC algorithms

To extract information from the posterior, MCMC algorithms are powerful and popular tools (Robert and Casella 2004; Brooks et al. 2011). In this work, we will consider the Metropolis-Adjusted Langevin Algorithm (MALA) (Roberts and Tweedie 1996), 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 is the overdamped Langevin stochastic differential equation (SDE):

$$d\boldsymbol{\theta} = \nabla \log \pi(\boldsymbol{\theta}|\mathbf{y})dt + \sqrt{2}dW, \tag{18}$$

where W is a standard $d_{\boldsymbol{\theta}}$-dimensional Brownian motion. Under mild conditions on the posterior π (Robert and Casella 2004), (18) is ergodic and has π as its stationary distribution, so that the probability density function of $\boldsymbol{\theta}(t)$ tends to π as $t \to \infty$.

Algorithm 1 Metropolis-Adjusted Langevin Algorithm

Require: initial value $\boldsymbol{\theta}_0$, initial acceptance rate $\alpha_0 = 0$, number of samples N, initial time-step γ_0, posterior $\pi(\boldsymbol{\theta}|\mathbf{y})$, optimal acceptance rate α_{opt}

while $n < N$ **do**

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

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

3. Compute the acceptance probability

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

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

4. Generate $r \sim U[0, 1]$.

if $r < \alpha_n$ **then**

$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}'$

$\mathbb{I}_n = 1$

else

$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n$.

$\mathbb{I}_n = 0$

end if

5. Update the time-step $\gamma_{n+1} = \gamma_n(1 + \frac{\mathbb{I}_n - \alpha_{opt}}{n+1})$

end while

In practice (18) is discretised with a simple Euler-Maruyama method with a time step γ:

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n + \gamma \nabla \log \pi(\boldsymbol{\theta}|\mathbf{y}) + \sqrt{2\gamma}\xi_n, \tag{19}$$

with $\xi_n \sim \mathcal{N}(0, 1)$. Assuming that the dynamics of (19) 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 (Sanz-Serna 2014). This gives rise to MALA as described 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 Sect. 2.3, which enables us to easily implement the MALA algorithm. Note that in contrast since the true posterior involves the (analytical or numerical) solution u to the PDE (1a)-(1b), it is usually impossible to compute these gradients analytically and one needs to resort to their numerical approximation. The following Lemma gives the gradient of the different approximate posteriors. The proof can be found in Appendix C.

Lemma 1 *Given the Gaussian process $\mathcal{G}_X^N \sim GP(\mathbf{m}_N^{\mathcal{G}_X}(\boldsymbol{\theta}), K_N(\boldsymbol{\theta}, \boldsymbol{\theta}))$ that emulates the forward map $\mathcal{G}_X$ with data $\mathcal{G}_X(\Theta)$, we have 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

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

where

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

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[2] in terms of the number of spatial and parameter dimensions as well as for different types of forward models. A common theme in all our experiments is that the number of

[2] A number of additional numerical experiments can be found in Appendix D

Table 2 Symbols and notations used in numerical experiments

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 $\mathbf{x} \in X = \{\mathbf{x}_i\}_{i=1}^{d_f}, \boldsymbol{\theta} \in \Theta = \{\boldsymbol{\theta}^i\}_{i=1}^{N_f}$
$\bar{N}$	We use $N_g = N_f = \bar{N}$

training points N is small, as this would be the case in large-scale applications in practice where increasing the number of training points is often either very costly or infeasible. The number of training points N used will serve as a benchmark for comparing different methodologies. Throughout all our numerical experiments in Sects. 4.1-4.3 when comparing the different approaches we keep N fixed. The value of N is chosen in such a way to ensure that significant uncertainty remains present in the emulator, which is typically the case in applications. Alternatively, one could ask what number of training points for each model is needed to reach a certain accuracy, however as explained above, this is not the viewpoint taken here. Precise timings for each of the approaches are reported in Sect. 4.4.

In cases where the PDE solution is not available in closed form, we use the finite element software Firedrake (Rathgeber et al. 2016) to obtain the "true" solution. Furthermore, when using MALA we adaptively tune the step size to achieve an average acceptance probability 0.573 (Brooks et al. 2011). In all our numerical experiments, we replace the uniform prior with a smooth approximation given by the $\lambda-$Moreau-Yosida envelope (Bauschke et al. 2011) with $\lambda = 10^{-3}$.

The selection of hyperparameters is crucial for the application of Gaussian process regression. In this paper, we optimize the hyperparameters by minimizing the negative log marginal likelihood, which in general could be computationally intensive. We simplify this process by assuming isotropy for the length-scale in the covariance function for $\boldsymbol{\theta}$ as in (5) and (6), so the optimization of the hyperparameters becomes a two-dimensional problem. Additionally, since we are operating in the small training data regime, the computational cost of evaluating the log-likelihood is small. To emphasise the improvement brought by the structure and also for simplicity, in the case of the spatially correlated and the PDE-constrained models, we use the same hyperparameters for the covariance function of the unknown parameter $\boldsymbol{\theta}$ as in the baseline model and only optimise the hyperparameters of the spatial covariance function. In principle, these assumptions can be relaxed to achieve potentially higher accuracy in the regression. The computational timings for optimization of the hyperparameters can be found in Appendix D.

To clarify the notation we use in our numerical experiments, we recall some of it in Table 2.

4.1 Examples in one spatial dimension

4.1.1 Two-dimensional piece-wise constant diffusion coefficient

We consider an elliptic equation with a 2-dimensional piece-wise constant diffusion coefficient; we have the following equation

$$\begin{aligned} &-\frac{\mathrm{d}}{\mathrm{d}x}(\exp(\kappa(x,\boldsymbol{\theta}))\frac{\mathrm{d}}{\mathrm{d}x}u(x)) = 4x, \\ &x \in (0,1), \quad \boldsymbol{\theta} \in [-1,1]^2, \\ &u(0) = 0, \qquad u(1) = 2, \end{aligned} \tag{20}$$

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

$$\kappa(x,\boldsymbol{\theta}) = \begin{cases} 0, & \text{for } x \in [0, \frac{1}{4}) \\ \theta_1, & \text{for } x \in [\frac{1}{4}, \frac{1}{2}) \\ \theta_2, & \text{for } x \in [\frac{1}{2}, \frac{3}{4}) \\ 1 & \text{for } x \in [\frac{3}{4}, 1] \end{cases} \tag{21}$$

Since it is not possible to solve (20) explicitly, we use Firedrake to obtain its solution.

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 $\boldsymbol{\theta}^\dagger = [0.098, 0.430]$, $d_\mathbf{y} = 6$ (equally spaced points in [0,1]) and noise level $\sigma_\eta^2 = 10^{-4}$. For the covariance kernels, we choose k_p to be the squared exponential kernel and k_s to be the Matèrn kernel with $\nu = \frac{5}{2}$.

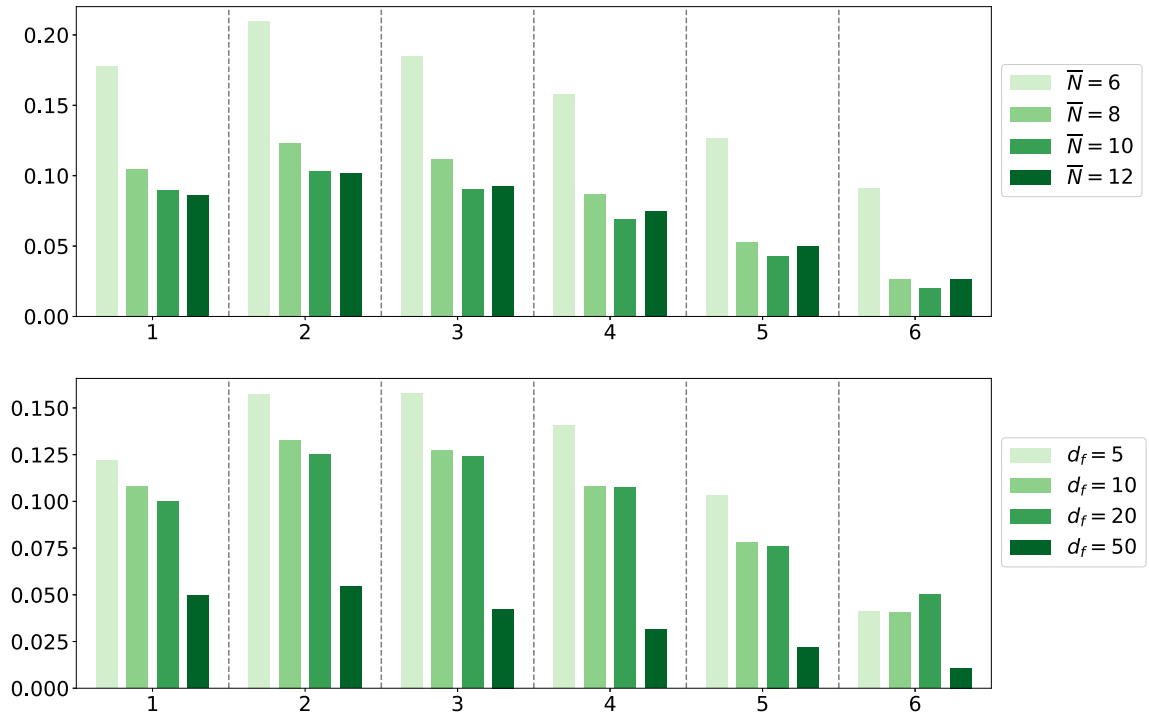

Fig. 1 Error between the predictive mean of PDE constrained emulator and the ground truth ($\boldsymbol{\theta} = \boldsymbol{\theta}^\dagger$) at the $d_{\mathbf{y}} = 6$ observation points for different $\bar{N}$ ($d_f = 20$) (*top plot*) and different d_f ($\bar{N} = 10$) (*bottom plot*) with $d_g = 2$ fixed

For the PDE constrained model, we first test the effect of additional training data $g(\Theta_g, X_g)$ and $f(\Theta_f, X_f)$ on the accuracy of the emulator. In Fig. 1, we see that as d_f and $\bar{N}$ increase, the accuracy of emulators gradually increases.

We now use MALA to obtain samples for all our approximate posteriors using 10^6 samples. For all models, 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 next 10 points in the Halton sequence), $d_f = 20$ and $d_g = 2$. Since we do not have access to the true posterior, we consider the results obtained from a mean-based approximation with the baseline model for $N = 10^2$ training points as the ground truth.

As we can see in Fig. 2, all the mean-based posteriors fail 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 approximations. 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 Fig. 3 where we plot the θ_1 and θ_2 marginals for all the mean-based posterior approximations $\pi^{N,\mathcal{G}_X}_{\text{mean}}$, $\pi^{N,\mathcal{G}_X,s}_{\text{mean}}$, $\pi^{N,\mathcal{G}_X,\text{PDE}}_{\text{mean}}$ and the marginal-based posterior approximations $\pi^{N,\mathcal{G}_X}_{\text{marginal}}$, $\pi^{N,\mathcal{G}_X,s}_{\text{marginal}}$, $\pi^{N,\mathcal{G}_X,\text{PDE}}_{\text{marginal}}$. Note that the marginal plot could be misleading regarding the overall performance of the approximations, for example in Fig. 3 (top right) the baseline model seems to be better than the PDE-constrained model, but from Fig. 2 we know that this is not true. In other words, the marginal posteriors are better approximations than the joint posterior. When we increase d_f from 20 to 50, the accuracy of the approximation improves as we can see in Fig. 4 where we compare PDE-constrained approximations for the two different values of d_f.

4.1.2 Parametric expansion for the diffusion coefficient

In this example, we study again (20), but this time instead of working with a piece-wise constant diffusion coefficient we assume that the diffusion coefficient satisfies the following parametric expansion

$$\kappa(\boldsymbol{\theta}, x) = \sum_{n=1}^{d_{\boldsymbol{\theta}}} \sqrt{a_n}\theta_n b_n(x) \tag{22}$$

where $a_n = \frac{8}{\omega_n^2+16}$, $b_n(x) = A_n(\sin(\omega_n x) + \frac{\omega_n}{4}\cos(\omega_n x))$, 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 $\|b_n\|_{L^2(0,1)} = 1$. This choice is motivated by the fact that for $\{\theta_n\}_{n=1}^{d_{\boldsymbol{\theta}}}$ i.i.d. standard normal random variables, this is a truncated Karhunen-Loève expansion of $\log(\kappa(\boldsymbol{\theta}, x)) \sim \text{GP}(0, \exp(-\|x - x'\|_1))$ (Ghanem and Spanos 1991).

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

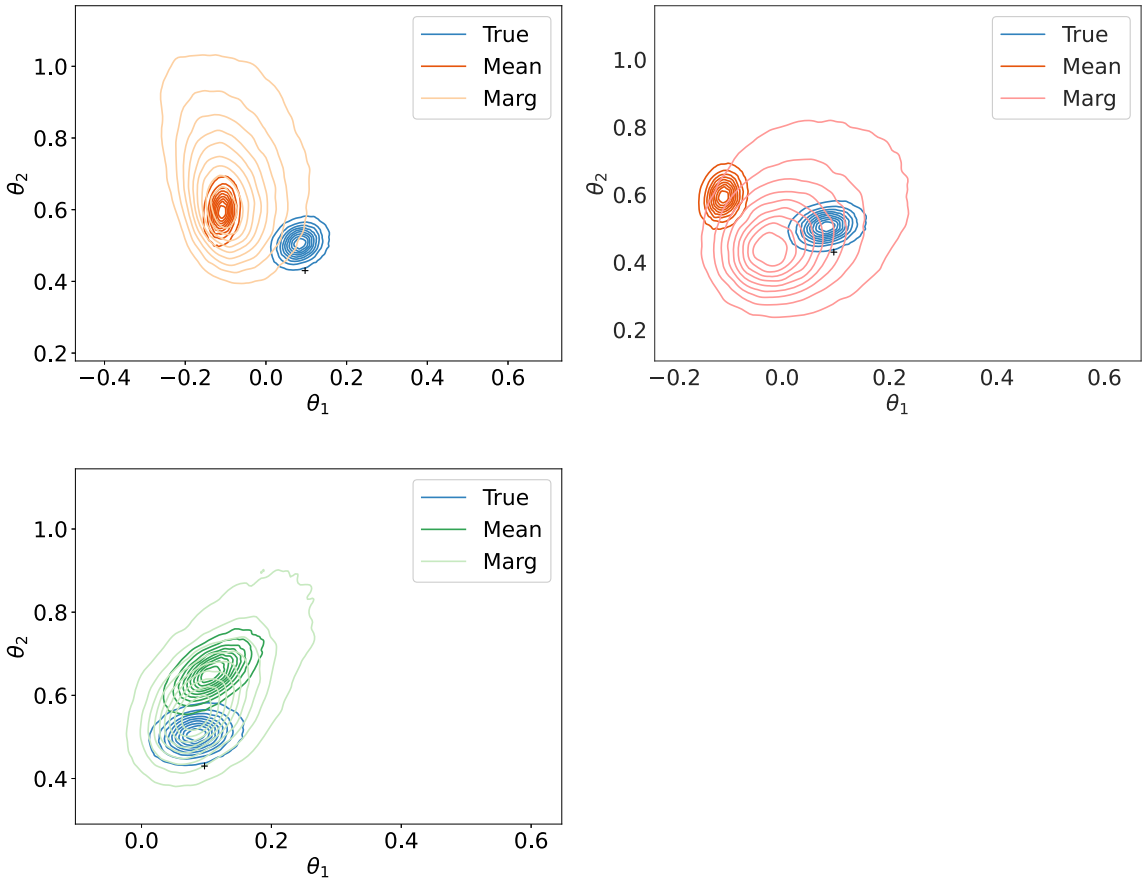

Fig. 2 Contour plots of the approximate mean-based and marginal-based posteriors: baseline model (*top left plot*), spatially correlated (*top right plot*), PDE-constrained (*bottom plot*). The symbol " + " denotes $\boldsymbol{\theta}^\dagger$. $\mathcal{G}_X$ is the discretised solution u in (20)

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 $[-1, 1]^2$. 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 Fig. 5 the θ_1 and θ_2 marginals for the different Gaussian emulators both in the case of mean-based and marginal posterior approximations. As we can see in Fig. 5 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 Fig. 5 (bottom left and bottom right) we can see that the marginals for the baseline and spatially correlated models 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.

4.2 Two spatial dimensions

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 first example, but depending only on the first dimension of $\mathbf{x}$:

$$\kappa(\mathbf{x}, \boldsymbol{\theta}) = \begin{cases} 0, & \text{for } x_1 \in [0, \frac{1}{4}), \\ \theta_1, & \text{for } x_1 \in [\frac{1}{4}, \frac{1}{2}), \\ \theta_2, & \text{for } x_1 \in [\frac{1}{2}, \frac{3}{4}), \\ 1, & \text{for } x_1 \in [\frac{3}{4}, 1]. \end{cases} \tag{23}$$

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

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

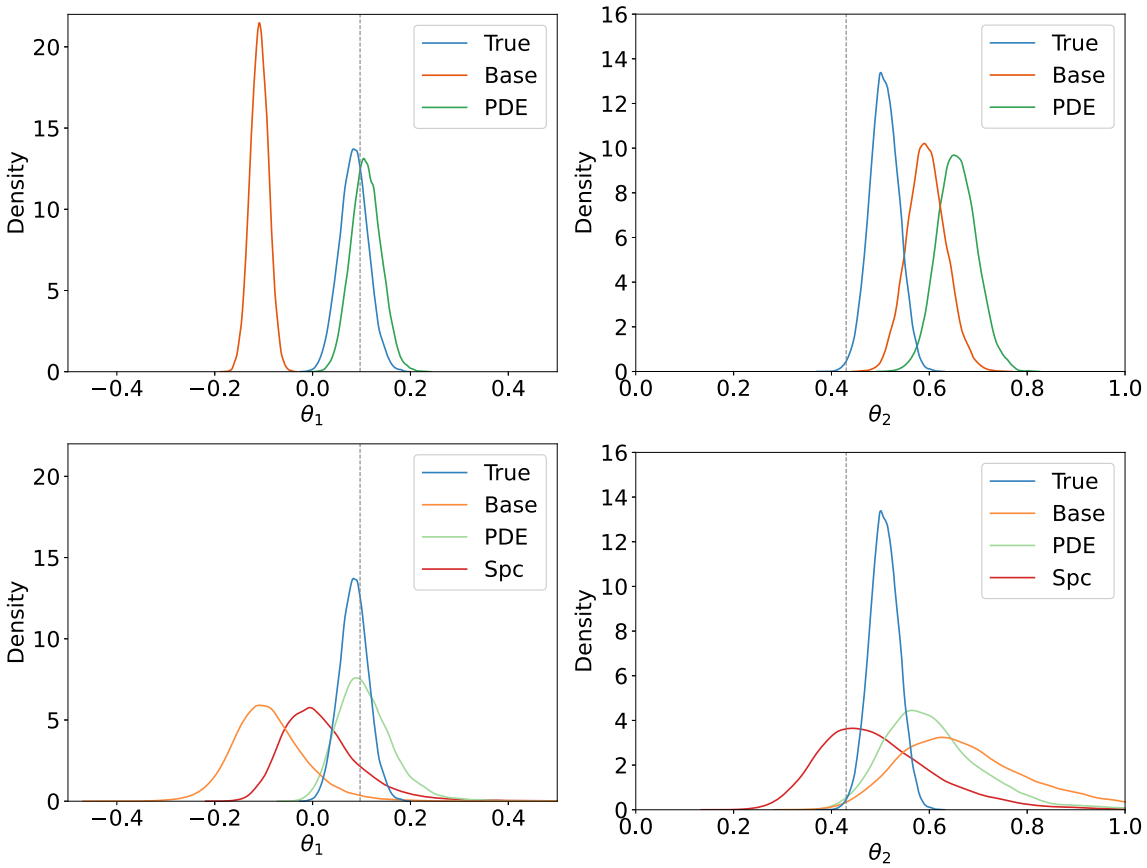

Fig. 3 Comparison of different models' marginal distribution when $N = 4$, for PDE model $d_f = 20$ and $\bar{N} = 20$: mean-based approximation of the θ_1 marginal (*top left plot*) and θ_2 marginal (*top right plot*), marginal approximation of the θ_1 marginal (*bottom left plot*) and θ_2 marginal (*bottom right plot*). $\mathcal{G}_X$ is the discretised solution u in (20) with diffusion coefficient (21)

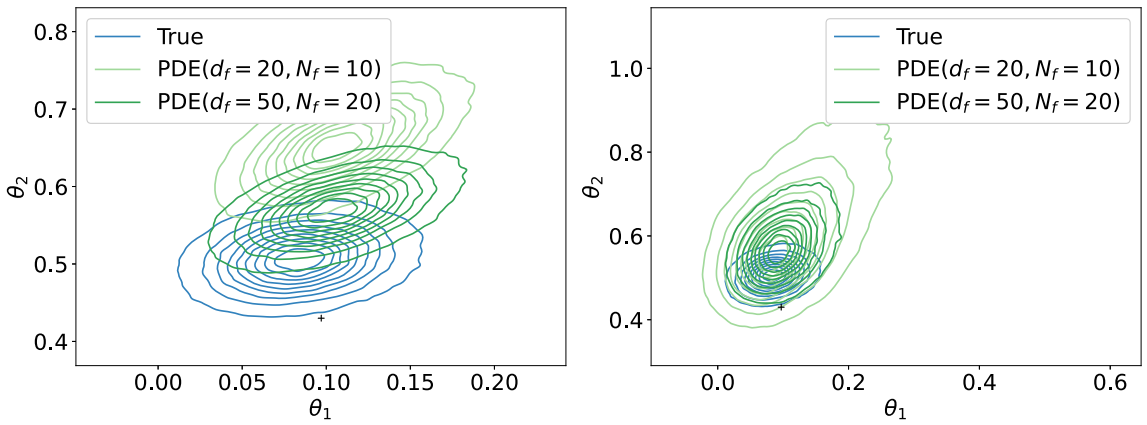

Fig. 4 Comparison of different models' marginal distribution when $N = 4$: mean-based approximation (*left plot*) and marginal-based approximation (*right plot*)

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 $x_1 = 0$.

For the observation, we generate our data $\mathbf{y}$ according to equation (2) for $\boldsymbol{\theta}^\dagger = [0.098, 0.430]$ with $d_{\mathbf{y}} = 6$ (chosen to be the first 6 points in the Halton sequence) and a noise level $\sigma_\eta^2 = 10^{-5}$. In addition, for the baseline and spatially correlated models, 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 $\nu = \frac{5}{2}$.

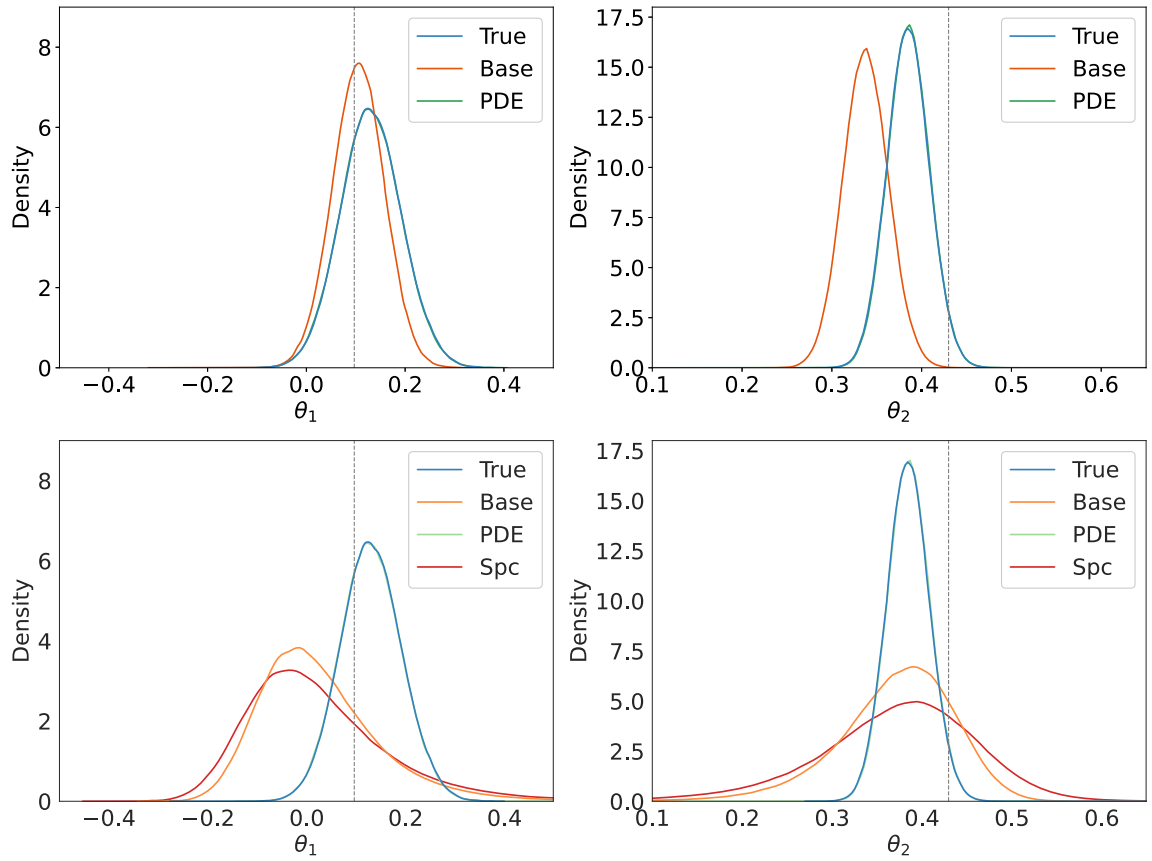

Fig. 5 Comparison of different models' marginal distribution when $N = 4$, for PDE model $\bar{N} = 10$ and $d_f = 8$: mean-based approximation of the θ_1 marginal (*top left plot*) and θ_2 marginal (*top right plot*), marginal approximation of the θ_1 marginal (*bottom left plot*) and θ_2 marginal (*bottom right plot*). $\mathcal{G}_X$ is the discretised solution u in (20) with diffusion coefficient (22) and $d_{\boldsymbol{\theta}} = 2$

We plot the mean-based approximate marginal posteriors in Fig. 6 (top left and top right). We can see that in this case, the PDE-constrained model significantly improves the approximation accuracy, which is different from the previous piece-wise constant diffusion coefficient example in one spatial dimension. In Fig. 6 (bottom left and bottom right), we compare the marginal approximation for the three models, and we again see that the PDE-constrained model performs best.

4.3 Emulating the negative log-likelihood

As discussed in Sect. 2.3.2, we can emulate the negative log-likelihood directly with Gaussian process regression instead of emulating the forward map. Since emulation of the log-likelihood involves emulating a non-linear functional of the PDE solution u, we are not able to incorporate spatial correlation or PDE constraints in the same way. We test the performance of the mean-based approximation (12) and the marginal approximation (13) using the previous examples: problem (20) with diffusion coefficient (21) with $d_{\mathbf{x}} = 1$ and $d_{\mathbf{x}} = 2$. All parameters are kept the same as in Sect. 4.1.1 and Sect. 4.2.

In Fig. 7, we compare the mean-based approximation with the 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 $d_{\mathbf{x}} = 1$, emulating the log-likelihood function performs better than the emulation of the observation with the baseline model, the approximated posterior is closer to the true posterior. For the $d_{\mathbf{x}} = 2$, its performance is much worse. Hence, emulating the log-likelihood with a small amount of data could be less reliable compared to emulating the forward map. If we increase the number of training data to $N = 10$ for the $d_{\mathbf{x}} = 2$ case, we can see an improvement of accuracy in Fig. 8, but it is still worse than emulating the forward map with the baseline model.

Similarly, we see in Fig. 9 that marginal approximations of the posterior based emulation of the log-likelihood appear to be less reliable, but including more training points can again improve the performance.

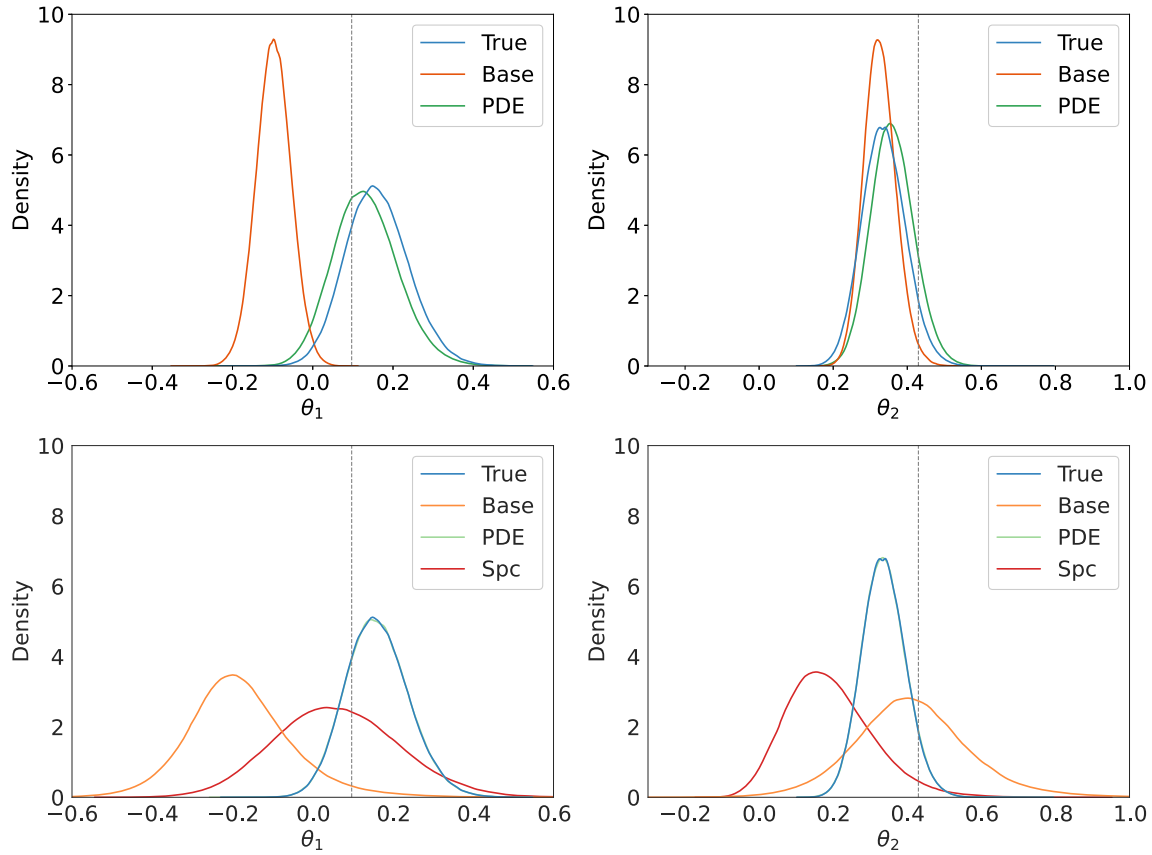

Fig. 6 Comparison of different models' marginal distribution when $N = 4$, for PDE model $\bar{N} = 30$ and $d_f = 30$: mean-based approximation of the θ_1 marginal (*top left plot*) and the θ_2 marginal (*top right plot*), and marginal approximation of the θ_1 marginal (*bottom left plot*) and the θ_2 marginal (*bottom right plot*). $\mathcal{G}_X$ is the discretised solution u with $d_{\mathbf{x}} = 2$ and diffusion coefficient (23)

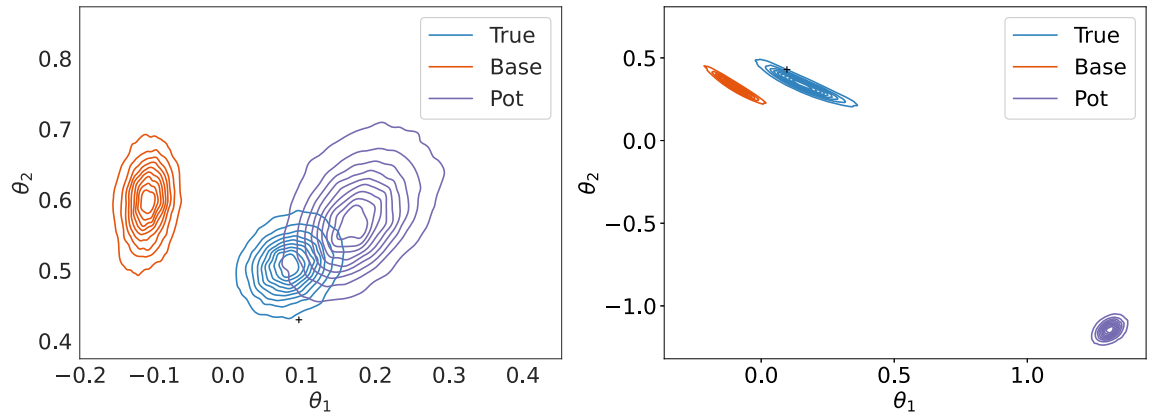

Fig. 7 Comparison of emulating log-likelihood function and emulating observations when $N = 4$. Both approximations are the mean-based approximation. $\mathcal{G}_X$ is the negative log-likelihood function in the problem (20) with the diffusion coefficient (21) with $d_{\mathbf{x}} = 1$ (*left plot*) and $d_{\mathbf{x}} = 2$ (*right plot*)

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, 6 and 7).

Table 3 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

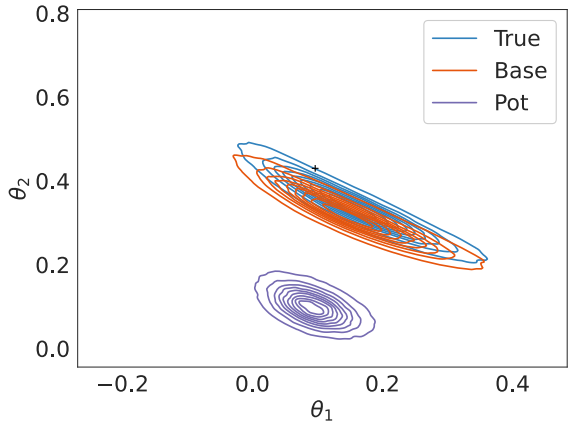

Fig. 8 The accuracy of the emulator is improved when N increases ($N = 10$). $\mathcal{G}_X$ is the negative log-likelihood function in the problem (20) with the diffusion coefficient (21) with $d_{\mathbf{x}} = 2$ and mean-based approximation

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 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 4, 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, whereas, 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.

Finally, Table 7 shows the effective sample sizes (ESSs) obtained for the different posterior approximations with MALA. We can see that the ESSs are all comparable, implying that it is meaningful to look at the cost per sample to compare the different approximate posteriors in terms of computational cost.

5 Conclusions, discussion and actionable advice

Bayesian inverse problems for PDEs pose significant computational challenges. The 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 within the 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.

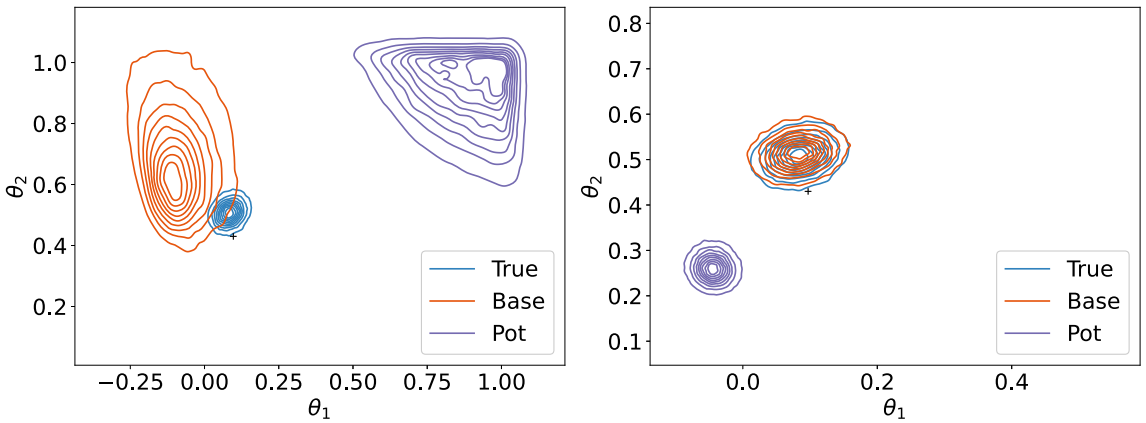

Fig. 9 Marginal approximation with $N = 4$ (*left plot*) and marginal approximation with $N = 10$ (*right plot*). $\mathcal{G}_X$ is the negative log-likelihood function in the problem (20) with the diffusion coefficient (21) with $d_{\mathbf{x}} = 1$

Table 3 Timings of PDE solution versus baseline Gaussian process emulator

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

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

Set-up	$\pi_{\text{marginal}}^{N,\mathcal{G}_X}$	$\pi_{\text{marginal}}^{N,\mathcal{G}_X,s}$	$\pi_{\text{marginal}}^{N,\mathcal{G}_X,\text{PDE}}$
$d_{\boldsymbol{\theta}} = 2, d_{\mathbf{y}} = 6, D = (0, 1), N = 4$	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 6 Timings of mean-based and marginal approximation when emulating the log-likelihood

Set-up	$\pi_{\text{mean}}^{N,\Phi}$	$\pi_{\text{marginal}}^{N,\mathcal{G}_X,\Phi}$
$d_{\boldsymbol{\theta}} = 2, d_{\mathbf{y}} = 6, D = (0, 1), N = 4$	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 7 Effective sample size for 10^6 samples

Model	Ess
Baseline mean	7274
Baseline marginal	10,337
Spatially correlated marginal	9249
PDE-constrained mean	9637
PDE-constrained marginal	9982

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 of 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 are 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 times 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 Sect. 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 the baseline model. The PDE-constrained model significantly increases the computational times for both the mean-based and marginal approximations, with the extent depending on the size of the 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 the spatially correlated/ PDE-constrained prior distributions provide mechanisms for 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. (Conrad et al. 2017; Calvetti et al. 2018; Fox et al. 2020). Conceptually, it is also related to including model discrepancy (Kennedy and O'Hagan 2000; Brynjarsdóttir and O'Hagan 2014). 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 hyperparameter 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 apply optimal experimental design in this work, i.e. how we should optimally choose the locations Θ of the training data. One would expect that using optimal design will have a large influence on the accuracy of the approximate posteriors, especially for small N. In the context of inverse problems, one usually wants to place the training points in regions of parameter space where the (approximate) posterior places significant mass (see e.g. Helin et al. (2023) 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.

A Derivation of marginal likelihood

Let $\mathbf{m}_{\boldsymbol{\theta}} = \mathbf{m}_N^{\mathcal{G}_X}(\boldsymbol{\theta})$, $K_{\boldsymbol{\theta}} = K_N(\boldsymbol{\theta}, \boldsymbol{\theta})$ and $\Gamma_\eta = \sigma_\eta^{-2} I_{d_y}$. Since $\mathcal{G}_X^N(\boldsymbol{\theta}) = \mathbf{m}_{\boldsymbol{\theta}} + \boldsymbol{\xi}$, where $\boldsymbol{\xi} \sim \mathcal{N}(0, K_{\boldsymbol{\theta}})$, using the definition of the expectation we obtain

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

We then rewrite and simplify the exponent part in the formula. Let $\bar{\mathbf{y}} = \mathbf{y} - \mathbf{m}_{\boldsymbol{\theta}}$, then

$$-\frac{1}{2}\left(\|\boldsymbol{\xi} - (\mathbf{y} - \mathbf{m}_{\boldsymbol{\theta}})\|_{\Gamma_\eta}^2 + \|\boldsymbol{\xi}\|_{K_{\boldsymbol{\theta}}}^2\right)$$
$$= -\frac{1}{2}\left(\|\boldsymbol{\xi} - \bar{\mathbf{y}}\|_{\Gamma_\eta}^2 + \|\boldsymbol{\xi}\|_{K_{\boldsymbol{\theta}}}^2\right)$$
$$= -\frac{1}{2}\left((\boldsymbol{\xi} - \bar{\mathbf{y}})^T\Gamma_\eta^{-1}(\boldsymbol{\xi} - \bar{\mathbf{y}}) + \boldsymbol{\xi}^T K_{\boldsymbol{\theta}}^{-1}\boldsymbol{\xi}\right)$$
$$= -\frac{1}{2}\left(\boldsymbol{\xi}^T(\Gamma_\eta^{-1} + K_{\boldsymbol{\theta}}^{-1})\boldsymbol{\xi}. - 2\bar{\mathbf{y}}^T\Gamma_\eta^{-1}\boldsymbol{\xi} + \bar{\mathbf{y}}^T\Gamma_\eta^{-1}\bar{\mathbf{y}}\right)$$

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

$$= -\frac{1}{2}\left(\boldsymbol{\xi}^T C^{-1} \boldsymbol{\xi} - 2\tilde{\mathbf{y}}^T C^{-1} \boldsymbol{\xi} + \bar{\mathbf{y}}^T \Gamma_\eta^{-1} \bar{\mathbf{y}}\right)$$

We can then complete the square

$$\begin{aligned}
&= -\frac{1}{2}\left(\|\boldsymbol{\xi} - \tilde{\mathbf{y}}\|_C^2 - \tilde{\mathbf{y}}^T C^{-1} \tilde{\mathbf{y}} + \bar{\mathbf{y}}^T \Gamma_\eta^{-1} \bar{\mathbf{y}}\right) \\
&= -\frac{1}{2}\left(\|\boldsymbol{\xi} - \tilde{\mathbf{y}}\|_C^2 - (C\Gamma_\eta^{-1}\bar{\mathbf{y}})^T C^{-1}(C\Gamma_\eta^{-1}\bar{\mathbf{y}}) + \bar{\mathbf{y}}^T \Gamma_\eta^{-1} \bar{\mathbf{y}}\right) \\
&= -\frac{1}{2}\left(\|\boldsymbol{\xi} - \tilde{\mathbf{y}}\|_C^2 - \bar{\mathbf{y}}^T \Gamma_\eta^{-1} K_{\boldsymbol{\theta}}(K_{\boldsymbol{\theta}} + \Gamma_\eta)^{-1} \bar{\mathbf{y}}) + \bar{\mathbf{y}}^T \Gamma_\eta^{-1} \bar{\mathbf{y}}\right) \\
&= -\frac{1}{2}\left(\|\boldsymbol{\xi} - \tilde{\mathbf{y}}\|_C^2 - \bar{\mathbf{y}}^T (\Gamma_\eta^{-1} K_{\boldsymbol{\theta}}(K_{\boldsymbol{\theta}} + \Gamma_\eta)^{-1} - \Gamma_\eta^{-1}) \bar{\mathbf{y}})\right) \\
&= -\frac{1}{2}\left(\|\boldsymbol{\xi} - \tilde{\mathbf{y}}\|_C^2 + \bar{\mathbf{y}}^T (K_{\boldsymbol{\theta}} + \Gamma_\eta)^{-1} \bar{\mathbf{y}}\right) \\
&= -\frac{1}{2}\|\bar{\mathbf{y}}\|^2_{(K_{\boldsymbol{\theta}} + \Gamma_\eta)} - \frac{1}{2}\left(\|\boldsymbol{\xi} - \tilde{\mathbf{y}}\|_C^2\right)
\end{aligned}$$

We now factor out $\exp\left(-\frac{1}{2}\left(\|\boldsymbol{\xi} - \tilde{\mathbf{y}}\|_C^2\right)\right)$ from the integral, and the remaining part matches the form of Gaussian distribution up to a constant.

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

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

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

B Predictive Gaussian process

B.1 Derivation

Gaussian prior given by (15)

Conditioning the Gaussian process prior (15) on the data $\mathcal{G}_X(\Theta)$ yields

$$\begin{aligned}
&\mathcal{G}_X(\boldsymbol{\theta})|\mathcal{G}_X(\Theta) \sim \mathrm{GP}(\mathbf{m}_N^{\mathcal{G}_X}(\boldsymbol{\theta}), K_N(\boldsymbol{\theta}, \boldsymbol{\theta}')), \\
&\text{with} \\
&\mathbf{m}_N^{\mathcal{G}_X}(\boldsymbol{\theta}) = K_{uu}(\boldsymbol{\theta}, \Theta) K_{uu}(\Theta, \Theta)^{-1} \mathcal{G}_X(\Theta), \\
&K_N(\boldsymbol{\theta}, \boldsymbol{\theta}') = K(\boldsymbol{\theta}, \boldsymbol{\theta}') - \\
&\qquad\qquad K_{uu}(\boldsymbol{\theta}, \Theta) K_{uu}(\Theta, \Theta)^{-1} K(\boldsymbol{\theta}', \Theta)^T \\
&\text{and} \\
&K_{uu}(\Theta, \Theta) = \{k_p(\boldsymbol{\theta}^i, \boldsymbol{\theta}^j) K_s(X, X)\} \in \mathbb{R}^{Nd_{\mathbf{y}} \times Nd_{\mathbf{y}}}, \\
&K_{uu}(\boldsymbol{\theta}, \Theta) = \{k_p(\boldsymbol{\theta}, \boldsymbol{\theta}^j) K_s(X, X)\} \in \mathbb{R}^{d_{\mathbf{y}} \times Nd_{\mathbf{y}}}.
\end{aligned} \tag{24}$$

Gaussian prior given by (17) We can condition the joint Gaussian process prior (17) as in Sect. 2.2 on the 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

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

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

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

and

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

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

$$
\begin{aligned}
&\mathcal{G}_X(\boldsymbol{\theta})|\mathcal{G}_X(\Theta), g(\Theta,X_g), f(\Theta,X_f)\\
&\qquad\sim \mathrm{GP}(\mathbf{m}_{N,X_f,X_g}^{\mathcal{G}_X}(\boldsymbol{\theta}), K_{N,X_f,X_g}(\boldsymbol{\theta},\boldsymbol{\theta}')),
\end{aligned}
\tag{25}
$$

where

$$
\begin{aligned}
&\mathbf{m}_{N,X_f,X_g}^{\mathcal{G}_X}(\boldsymbol{\theta}) =\\
&\left[K_{uu}(\boldsymbol{\theta},\Theta),\ K_{ug}(\boldsymbol{\theta},\Theta),\ K_{uf}(\boldsymbol{\theta},\Theta)\right]\\
&\quad\times \tilde{K}(\Theta,\Theta)^{-1}\mathbf{g}(\Theta),\\
&K_{N,X_f,X_g}(\boldsymbol{\theta},\boldsymbol{\theta}') = K(\boldsymbol{\theta},\boldsymbol{\theta}')\\
&-\left[K_{uu}(\boldsymbol{\theta},\Theta),\ K_{ug}(\boldsymbol{\theta},\Theta),\ K_{uf}(\boldsymbol{\theta},\Theta)\right]\\
&\quad\times \tilde{K}(\Theta,\Theta)^{-1}\begin{bmatrix}K_{uu}(\boldsymbol{\theta}',\Theta)\\ K_{ug}(\boldsymbol{\theta}',\Theta)\\ K_{uf}(\boldsymbol{\theta}',\Theta)\end{bmatrix}.
\end{aligned}
$$

B.2 Computational implementation

We 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 2 below). This was already noted in Bonilla et al. (2007), but we give a proof for this for completeness. 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,

$$
\underbrace{K(\Theta_1,\Theta_2)}_{N_1d_{\mathbf{y}}\times N_2d_{\mathbf{y}}} = \underbrace{K_p(\Theta_1,\Theta_2)}_{(N_1\times N_2)} \otimes \underbrace{K_s(X,X)}_{(d_{\mathbf{y}}\times d_{\mathbf{y}})}. \tag{26}
$$

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

Theorem 2 *Consider two Gaussian processes* $\mathbf{g}_0(\boldsymbol{\theta}) \sim GP(\mathbf{m}(\boldsymbol{\theta}), k_p(\boldsymbol{\theta},\boldsymbol{\theta}')I_{d_{\mathbf{y}}})$, $\mathbf{g}_{0,s}(\boldsymbol{\theta}) \sim GP(\mathbf{m}(\boldsymbol{\theta}), k_p(\boldsymbol{\theta},\boldsymbol{\theta}')K_s(X,X))$, *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,*

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

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

Proof Let $k_p(\boldsymbol{\theta},\Theta) := [k_p(\boldsymbol{\theta},\boldsymbol{\theta}^1);\ldots;k_p(\boldsymbol{\theta},\boldsymbol{\theta}^N)] \in \mathbb{R}^{d_{\mathbf{y}}}$, 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

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

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

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

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 (16) also depends explicitly on the parameters $\boldsymbol{\theta}$. Therefore, its covariance matrix cannot be written in a Kronecker product structure as in (26) and Theorem 2 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.

C Derivation of the gradient of the approximate log-posteriors

For the mean-based posterior:

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

For the marginal posterior

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

where

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

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

D Further numerical experiments

D.1 Constant diffusion coefficient

We consider the following PDE in one spatial dimension

$$
\begin{aligned}
&-\frac{\mathrm{d}}{\mathrm{d}x}\left(e^\theta \frac{\mathrm{d}u(x)}{\mathrm{d}x}\right) = 1, \\
&x \in (0, 1), \quad \theta \in [-1, 1], \\
&u(0) = 0, \quad u(1) = 0.
\end{aligned} \tag{27}
$$

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, we calculate the true and approximate posteriors on a fine grid without having to resort to Markov Chain Monte Carlo sampling. We now generate our observations $\mathbf{y}$ according to equation (2) for $\boldsymbol{\theta}^\dagger = 0.314$ at a varying number of spatial points $d_{\mathbf{y}}$ (equally spaced in $[0, 1]$) and at a noise level $\sigma_\eta^2 = 10^{-5}$. As we can see in Fig. 10 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. Giordano and Nickl (2020) for related results).

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

Baseline model: In the case of the simplest emulator with independent entries, we illustrate in Fig. 11, 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 locations of the training

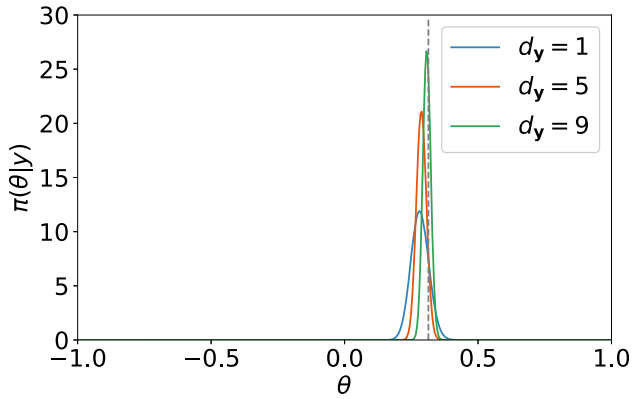

Fig. 10 The true posteriors with different d_y

points are chosen from the Halton sequence (Niederreiter 1992). Now, when comparing Fig. 11 (top left and top right) 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 $\boldsymbol{\theta}^\dagger$ and exhibits bimodality since it so happens that the training point used is not near the true $\boldsymbol{\theta}^\dagger$. However, due to the variance inflation this is not the case for the marginal-based posterior. Furthermore, in Fig. 11 (bottom left) 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 slightly smaller than the error for the mean-based posterior for small N. The two errors are equal as N increases, which can be further understood by Fig. 11(bottom right). Here, we plot the average variance $k_N(\boldsymbol{\theta}, \boldsymbol{\theta})$ (averaged over $\boldsymbol{\theta}$) of our emulator for different values of N, and see that as expected from (11) the marginal approximation behaves in the same manner as the mean-based approximation once the average variance is of the same order as the observational noise σ_η^2.

Spatially correlated model: As discussed in Appendix B, the introduction of spatial correlation does not change the predictive mean of the Gaussian processes. We hence now compare in Fig. 12 the two different marginal posteriors $\pi^{N,\mathcal{G}_X}_{\text{marginal}}$ and $\pi^{N,\mathcal{G}_X,s}_{\text{marginal}}$, where the latter includes spatial correlation. We

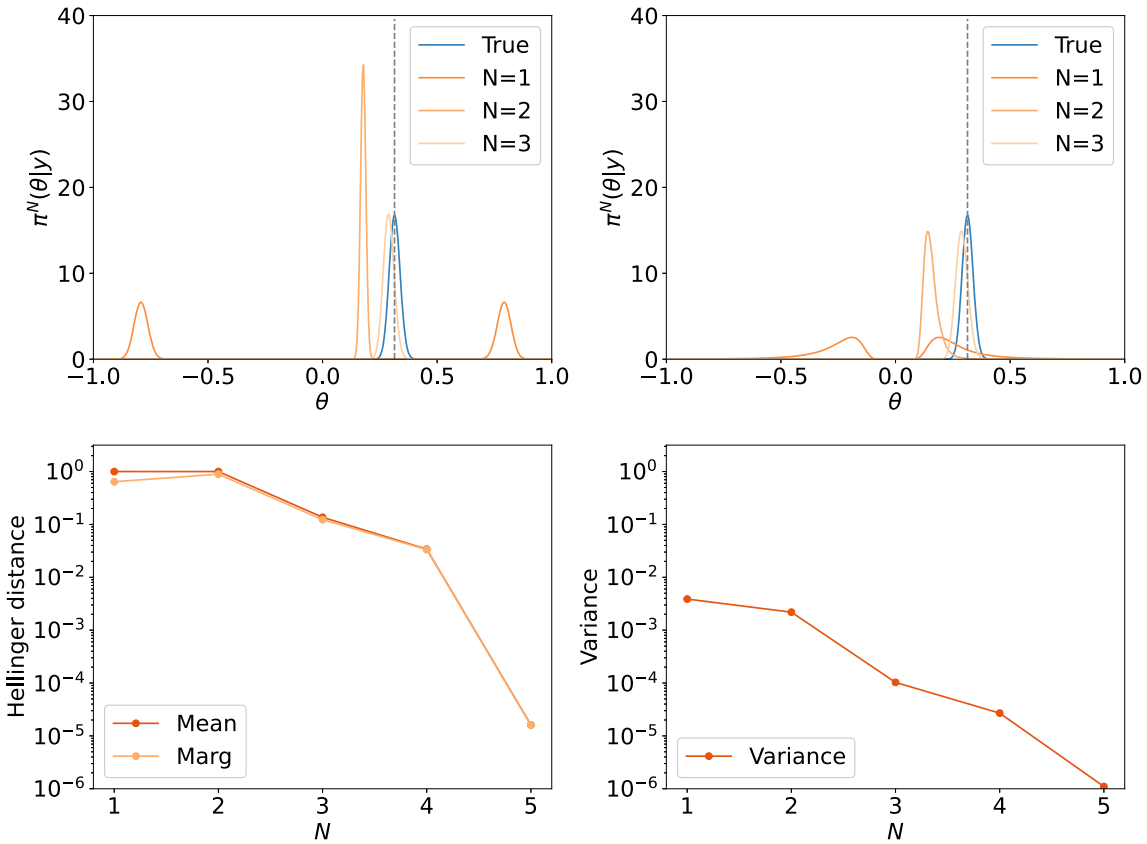

Fig. 11 Baseline model mean-based posterior (*top left*) and marginal posterior (*top right*) with different N. The Hellinger distance between approximated posteriors and the true posterior (*bottom left*) and average predictive variance of the Gaussian process emulator (*bottom right*) as N increases. $\mathcal{G}_X$ is the discretised solution u in (27)

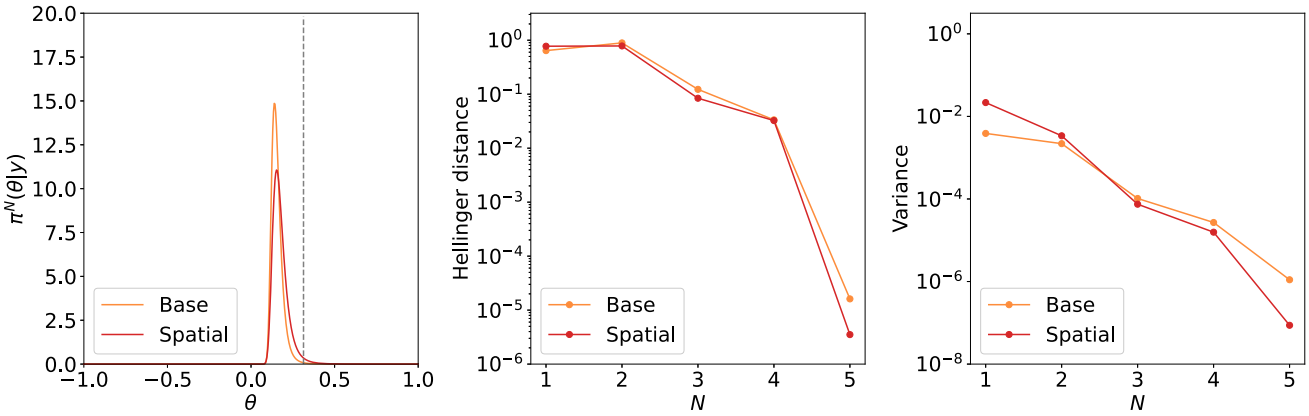

Fig. 12 Baseline and spatially correlated model marginal posteriors for $N = 2$ (*left plot*). The Hellinger distance between approximated posteriors and the true posterior (*middle plot*) and average predictive variance of the Gaussian process emulator (*right plot*) as N increases. $\mathcal{G}_X$ is the discretised solution u in (27)

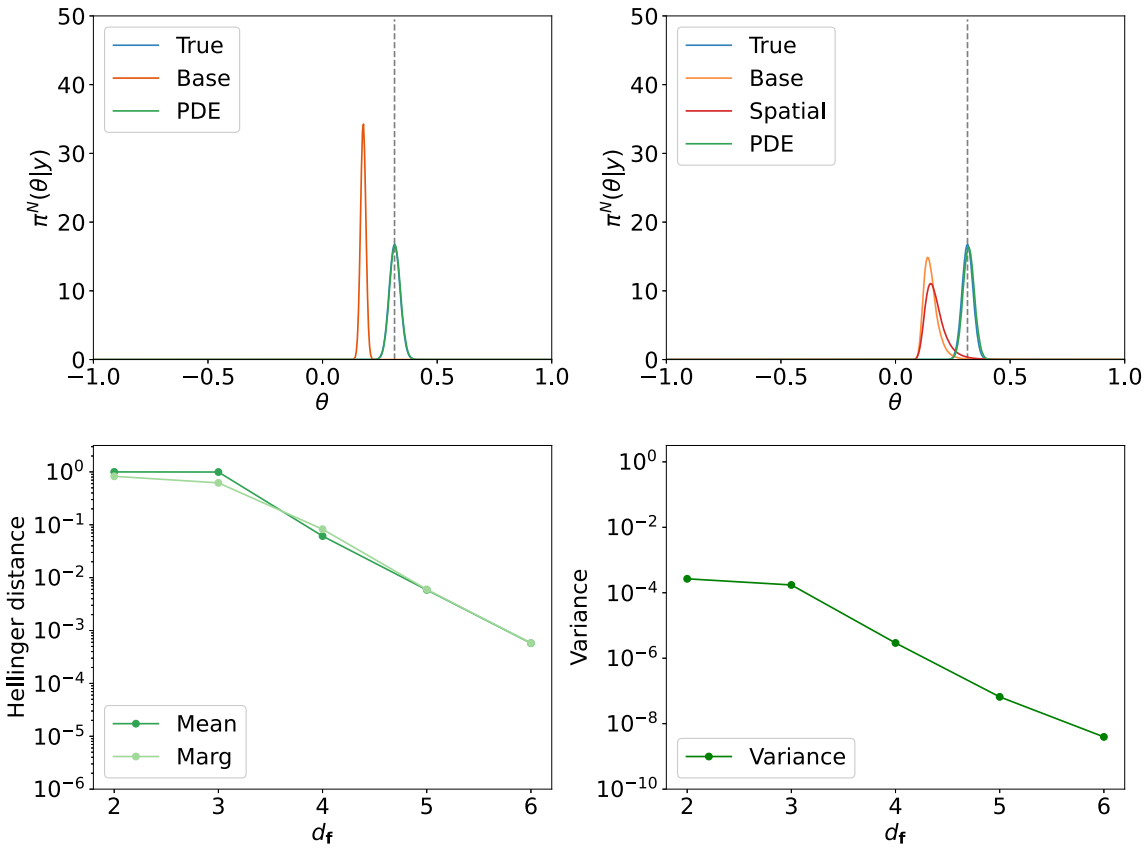

Fig. 13 Comparison of different models when $N = 2$, for PDE model $d_f = 5$: mean-based posteriors (*top left plot*) and marginal posteriors (*top right plot*). The Hellinger distance between approximated posteriors and the true posterior (*bottom left plot*) and average predictive variance of the emulator (*bottom right plot*) as d_f increases. $\mathcal{G}_X$ is the discretised solution u in (27)

again choose $d_{\mathbf{y}} = 5$. In particular, as we can see in Fig. 12 (left) (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 Fig. 12) leads to similar behavior as in Fig. 11 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 Fig. 12 (middle), as we increase the number of training points for our Gaussian process, the Hellinger distance between the

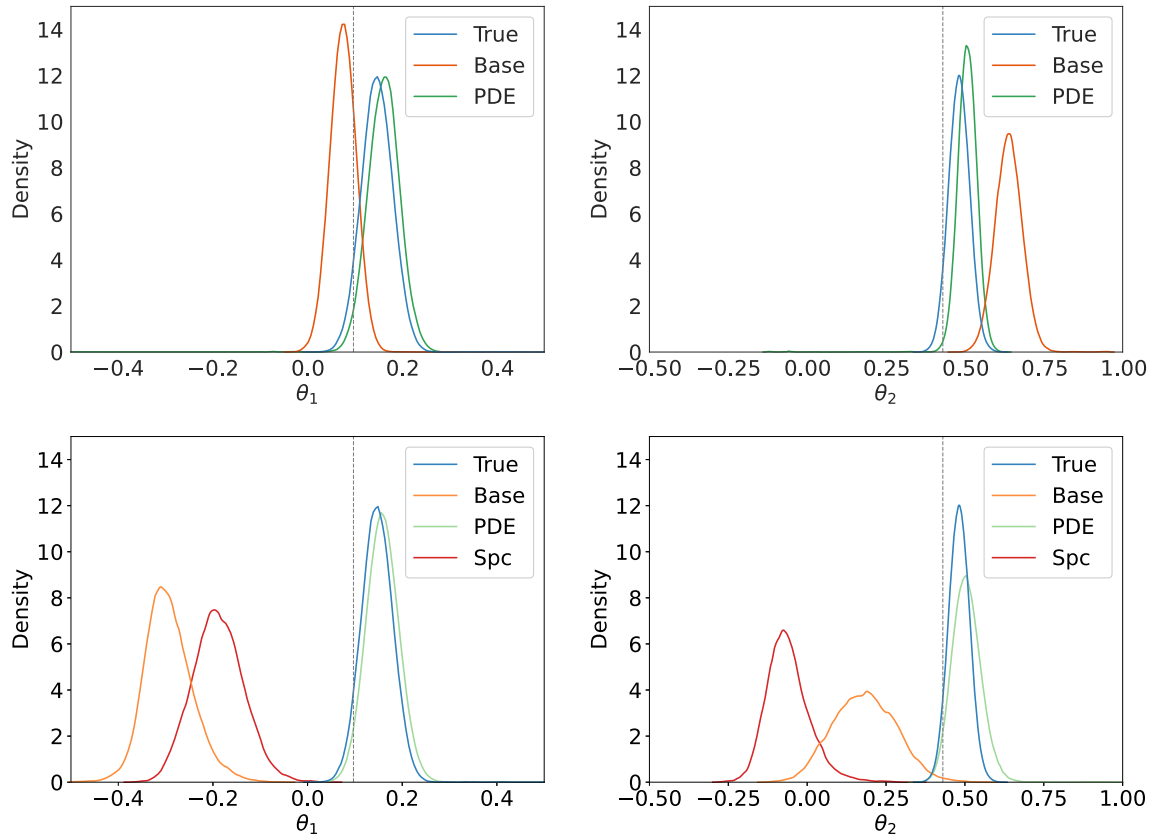

Fig. 14 Comparison of different models' marginal distribution when $N = 4$, for PDE model $\bar{N} = 10$ and $d_f = 50$: mean-based approximation of the θ_1 marginal (*top left plot*) and θ_2 marginal (*top right plot*), marginal approximation of the θ_1 marginal (*bottom left plot*) and θ_2 marginal (*bottom right plot*). $\mathcal{G}_X$ is the integrals of solution u in (20) with diffusion coefficient (21)

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 the marginal posterior (again here $d_{\mathbf{y}} = 5$). In particular, as we can see in Figs. 13 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 the 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 different ranges of values for N_f. It is also worth noting that when comparing with the Hellinger distance from Figs. 11 (bottom left) and 12 (middle) 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 Fig. 13 (bottom right), 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.

D.2 Integral observation operator

We now investigate the proposed method with a different form of the observation operator. In terms of the PDE problem, we study again (20). However, instead of pointwise 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 $\boldsymbol{\theta}^\dagger = [0.098, 0.430]$, $d_{\mathbf{y}} = 16$ (equally spaced sub-intervals of $[0, 1]$) and $\sigma_\eta^2 = 10^{-6}$. We do not conduct precise integration as in (27), but use MALA algorithm to obtain our samples. We utilize 10^6 samples for all our approximate posteriors. 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 Fig. 14, 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 simi-

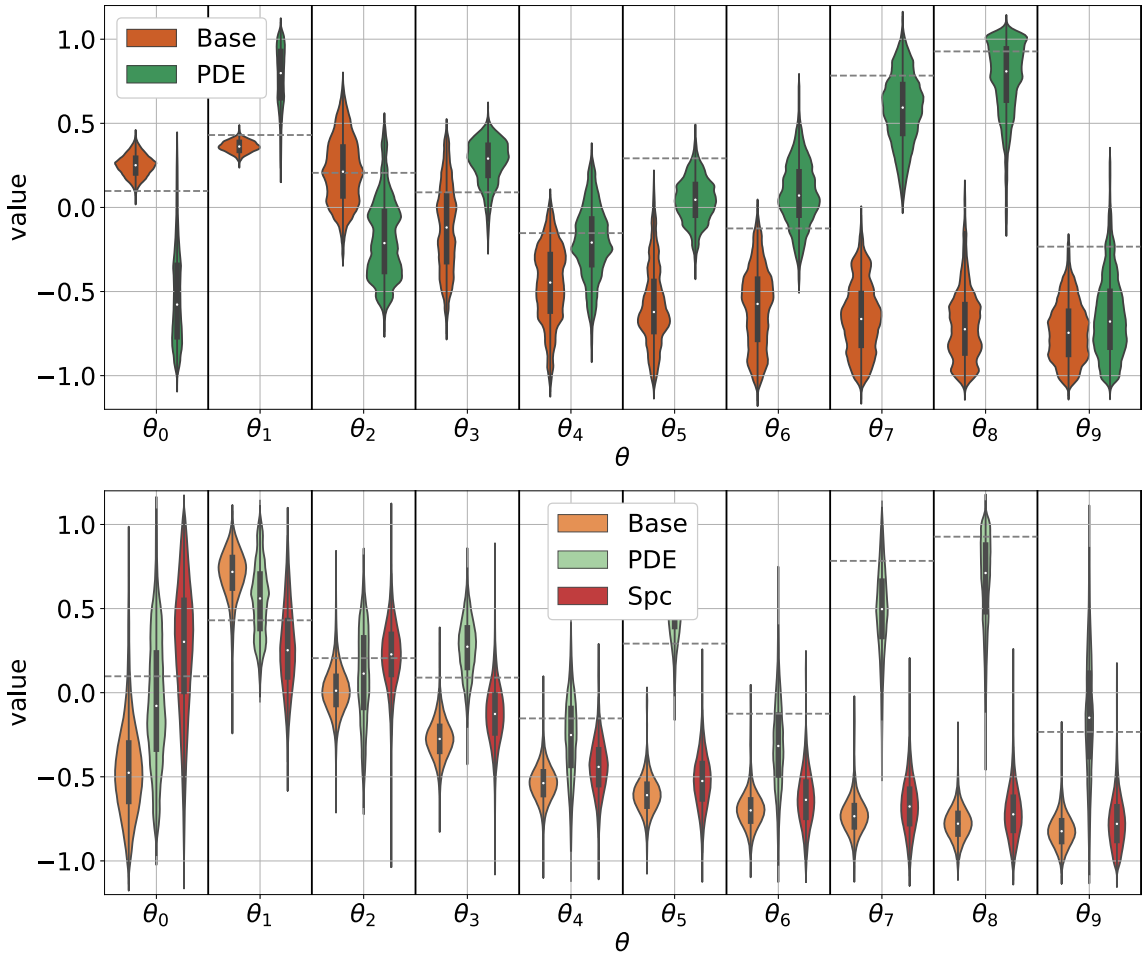

Fig. 15 Comparison of different models' marginal distribution when $N = 4$, for PDE model $\bar{N} = 50$ and $d_f = 25$: mean-based approximation (*top plot*) and marginal approximation (*bottom plot*). $\mathcal{G}_X$ is the discretised solution u in (20) with diffusion coefficient (22) and $d_{\boldsymbol{\theta}} = 10$. The true parameter $\boldsymbol{\theta}^\dagger$ is being plotted as horizontal dashed lines

lar to the example in Sect. 4.2 that the PDE-constrained model performs better than the other two models.

D.3 Ten-dimensional parametric expansion diffusion coefficient

We now increase the dimension of the diffusion coefficient from $d_{\boldsymbol{\theta}} = 2$ to $d_{\boldsymbol{\theta}} = 10$ in (22), to test the proposed method in a relatively high dimensional space. 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 the same as before ($\sigma_\eta^2 = 10^{-4}$). The number of training points for all emulators is again set to $N = 4$, 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 for 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 Fig. 15, 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 to the model helps the approximation move towards the true value, but it still cannot correctly approximate the posterior for the last few variables. The performance of the PDE-constrained

Table 8 Timings of hyperparameter optimization

Model	Time
Baseline model (σ, l for k_p)	$0.18s$
Spatially correlated model (l for k_s)	$0.04s$
PDE-constrained model (l for k_s)	$45s$

model is much better than that of the other models, it is placing the posterior mass around the true value for all variables.

D.4 Timing of hyperparameter optimization

Table 8 gives computational timings for computing the hyperparameters in the covariance functions. The optimization of the hyperparameters for k_s involves repeatedly computing the inverse of the Gram matrix. In the PDE-constrained model, since the matrix is significantly augmented by the data of PDE, the computation timing is therefore much longer than that of the other two models.

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, EP/V006177/1, EP/X01259X/1 and EP/Y028783/1.

Author Contributions All authors contributed equally.

Data availibility No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no competing interests.

References

Alvarez, M.A., Rosasco, L., Lawrence, N.D.: Kernels for vector-valued functions: a review. Foundations and Trends® in Machine Learning **4**(3), 195–266 (2012)

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

Bauschke, H.H., Burachik, R.S., Combettes, P.L., Elser, V., Luke, D.R., Wolkowicz, H.: Fixed-point Algorithms for Inverse Problems in Science and Engineering, vol. 49. Springer, New York, NY (2011)

Bonilla, E.V., Chai, K., Williams, C.: Multi-task Gaussian process prediction. Advances in neural information processing systems **20** (2007)

Brooks, S., Gelman, A., Jones, G., Meng, X.-L.: Handbook of Markov Chain Monte Carlo. CRC Press, Boca Raton, FL (2011)

Brynjarsdóttir, J., O'Hagan, A.: Learning about physical parameters: The importance of model discrepancy. Inverse Prob. **30**(11), 114007 (2014)

Bui-Thanh, T., Willcox, K., Ghattas, O.: Model reduction for large-scale systems with high-dimensional parametric input space. SIAM J. Sci. Comput. **30**(6), 3270–3288 (2008)

Calvetti, D., Dunlop, M., Somersalo, E., Stuart, A.: Iterative updating of model error for Bayesian inversion. Inverse Prob. **34**(2), 025008 (2018)

Cockayne, J., Oates, C., Sullivan, T., Girolami, M.: Probabilistic numerical methods for PDE-constrained Bayesian inverse problems. AIP Conf. Proc. **1853**(1), 060001 (2017)

Conrad, P.R., Girolami, M., Särkkä, S., Stuart, A., Zygalakis, K.: Statistical analysis of differential equations: introducing probability measures on numerical solutions. Stat. Comput. **27**, 1065–1082 (2017)

Constantine, P.G.: Active Subspaces. Society for Industrial and Applied Mathematics, Philadelphia, PA (2015)

Constantine, P.G., Dow, E., Wang, Q.: Active subspace methods in theory and practice: applications to kriging surfaces. SIAM J. Sci. Comput. **36**(4), 1500–1524 (2014)

Fox, C., Cui, T., Neumayer, M.: Randomized reduced forward models for efficient metropolis-hastings mcmc, with application to subsurface fluid flow and capacitance tomography. GEM-Int. J. Geomath. **11**, 1–38 (2020)

Ghanem, R.G., Spanos, P.D.: Stochastic Finite Elements: a Spectral Approach. Springer, Berlin (1991)

Giordano, M., Nickl, R.: Consistency of Bayesian inference with Gaussian process priors in an elliptic inverse problem. Inverse Prob. **36**(8), 085001 (2020)

Helin, T., Stuart, A.M., Teckentrup, A.L., Zygalakis, K.C.: 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)

Higdon, D., Kennedy, M., Cavendish, J.C., Cafeo, J.A., Ryne, R.D.: Combining field data and computer simulations for calibration and prediction. SIAM J. Sci. Comput. **26**(2), 448–466 (2004)

Kaipio, J., Somersalo, E.: Statistical and Computational Inverse Problems. Springer, Dordrecht (2005)

Kennedy, M.C., O'Hagan, A.: Bayesian calibration of computer models. J. R. Stat. Soc. Ser. B Methodol. **63**, 425–464 (2000)

Lie, H.C., Sullivan, T.J., Teckentrup, A.L.: Random forward models and log-likelihoods in Bayesian inverse problems. SIAM/ASA Journal on Uncertainty Quantification **6**(4), 1600–1629 (2018)

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

Marzouk, Y.M., Najm, H.N., Rahn, L.A.: Stochastic spectral methods for efficient Bayesian solution of inverse problems. J. Comput. Phys. **224**(2), 560–586 (2007)

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

Niederreiter, H.: Random Number Generation and quasi-Monte Carlo Methods. SIAM, London (1992)
Oates, Chris J., Jon Cockayne, R.G.A., Girolami, M.: Bayesian probabilistic numerical methods in time-dependent state estimation for industrial hydrocyclone equipment. J. Am. Stat. Assoc. **114**(528), 1518–1531 (2019)
O'Hagan, A.: Bayesian analysis of computer code outputs: a tutorial. Reliab. Eng. Syst. Saf. **91**(10), 1290–1300 (2006)
Pförtner, M., Steinwart, I., Hennig, P., Wenger, J.: Physics-informed Gaussian process regression generalizes linear PDE solvers (2022)
Raissi, M., Perdikaris, P., Karniadakis, G.E.: Machine learning of linear differential equations using Gaussian processes. J. Comput. Phys. **348**, 683–693 (2017)
Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. MIT Press, Cambridge, MA (2006)
Rathgeber, F., Ham, D.A., Mitchell, L., Lange, M., Luporini, F., McRae, A.T., Bercea, G.-T., Markall, G.R., Kelly, P.H.: Firedrake: automating the finite element method by composing abstractions. ACM Transactions on Mathematical Software (TOMS) **43**(3), 1–27 (2016)
Robert, C.P., Casella, G.: Monte Carlo Statistical Methods. Springer, Berlin (2004)
Roberts, G.O., Tweedie, R.L.: Exponential convergence of langevin distributions and their discrete approximations. Bernoulli, 341–363 (1996)
Sacks, J., Welch, W.J., Mitchell, T.J., Wynn, H.P.: Design and analysis of computer experiments. Stat. Sci. **4**(4), 409–423 (1989)
Sanz-Serna, J.M.: Markov Chain Monte Carlo and Numerical Differential Equations, 39–88. Springer, Cham (2014)
Spitieris, M., Steinsland, I.: Bayesian calibration of imperfect computer models using physics-informed priors. J. Mach. Learn. Res. **24**(108), 1–39 (2023)
Stein, M.L.: Interpolation of Spatial Data. Springer Series in Statistics, 247. Springer, New York, NY (1999)
Stuart, A.M.: Inverse problems: a Bayesian perspective. Acta Numer. **19**, 451–559 (2010)
Stuart, A., Teckentrup, A.: Posterior consistency for Gaussian process approximations of Bayesian posterior distributions. Math. Comput. **87**, 721–753 (2018)
Swiler, L.P., Gulian, M., Frankel, A.L., Safta, C., Jakeman, J.D.: Constrained Gaussian processes: A survey. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States) (2021)
Teckentrup, A.L.: Convergence of Gaussian process regression with estimated hyper-parameters and applications in Bayesian inverse problems. SIAM/ASA Journal on Uncertainty Quantification **8**(4), 1310–1337 (2020)
Xiu, D., Karniadakis, G.E.: Modeling uncertainty in flow simulations via generalized polynomial chaos. J. Comput. Phys. **187**(1), 137–167 (2003)