Deep Gaussian Processes for Calibration of Computer Models

Sébastien Marmin† and Maurizio Filippone‡

Abstract. Bayesian calibration of black-box computer models offers an established framework for quantification of uncertainty of model parameters and predictions. Traditional Bayesian calibration involves the emulation of the computer model and an additive model discrepancy term using Gaussian processes; inference is then carried out using Markov chain Monte Carlo. This calibration approach is limited by the poor scalability of Gaussian processes and by the need to specify a sensible covariance function to deal with the complexity of the computer model and the discrepancy. In this work, we propose a novel calibration framework, where these challenges are addressed by means of compositions of Gaussian processes into Deep Gaussian processes and scalable variational inference techniques. Thanks to this formulation, it is possible to obtain a flexible calibration approach, which is easy to implement in development environments featuring automatic differentiation and exploiting GPU-type hardware. We show how our proposal yields a powerful alternative to the state-of-the-art by means of experimental validations on various calibration problems. We conclude the paper by showing how we can carry out adaptive experimental design, and by discussing the identifiability properties of the proposed calibration model.

Keywords: Bayesian inference, neural nets, computer experiments, deep Gaussian process.


1 Introduction

The task of carrying out inference of parameters of expensive computer models from data is a classical problem in Statistics (Sacks et al., 1989). Such a problem is referred to as calibration (Kennedy and O’Hagan, 2001), and the results are often of interest to draw conclusions on parameters that have a direct interpretation of physical quantities (see, e.g., Section 2.2 of Brynjarsdóttir and O’Hagan 2014). Calibration finds numerous applications in fields as diverse as climatology (Sansó et al., 2008; Salter et al., 2018), environmental sciences (Larssen et al., 2006; Arhonditis et al., 2007), biology (Henderson et al., 2009), and mechanical engineering (Williams et al., 2006), to name a few. There are many fundamental difficulties in calibrating expensive computer models, which we can distinguish between computational and statistical. Computational issues

---

*MF gratefully acknowledges support from the AXA Research Fund and the Agence Nationale de la Recherche (grant ANR-18-CE46-0002 and ANR-19-P3IA-0002).
†Laboratoire national de métrologie et d’essais (LNE), 29 avenue Hennequin, 78197 Trappes, sebastien.marmin@lne.fr
‡EURECOM, Campus SophiaTech 450 Route des Chappes, 06410 Biot

© 2022 International Society for Bayesian Analysis https://doi.org/10.1214/21-BA1293
arise from the fact that traditional optimization and inference techniques require running the expensive computer model many times for different values of the parameters, which might be unfeasible within a given computational budget. Statistical limitations, instead, arise from the fact that computer models can only abstract real processes with a given level of accuracy.

Building on previous work from Sacks et al. (1989), in their seminal paper, Kennedy and O’Hagan (2001) propose a statistical model, based on Gaussian processes (GPs; Rasmussen and Williams 2006), which jointly tackles the problems above. In their model, which we will refer to as KOH, the output of a deterministic computer model is emulated through a GP estimated from a set of computer experiments; in this way, computational issues are bypassed by using the predictive distribution of the emulating GP for any given set of parameters in place of expensive runs of the computer model. Observations from the real process, also known as field observations, instead, are modeled through the GP emulating the computer model with the addition of a so-called discrepancy term, which is also assigned a GP prior. The introduction of the discrepancy term is key to avoid biased estimates of the parameters due to misspecifications of the computer model (Brynjarsdóttir and O’Hagan, 2014). The KOH model is treated in a Bayesian way, making it suitable for problems where quantification of uncertainty is an important requirement. This is often the case when one is interested in drawing conclusions on parameters of interest, making predictions for decision-making with specific cost associated to predictions, or when one is interested in iteratively improving the experimental design.

While the KOH model and inference make for an attractive and elegant framework to tackle quantification of uncertainty for calibration of expensive computer models, there are limitations which we aim to overcome with this work. From the modeling perspective, GPs are indeed flexible emulators, provided that a suitable covariance function is chosen, as in the literature of nonstationary GPs (e.g. Paciorek and Schervish 2003). However, more recent approaches like Deep GPs (DGP; Damianou and Lawrence 2013) have shown great modeling flexibility for many classes of functions and can potentially lead to more accuracy in the emulation of the computer model and the real process compared to GPs. From the computational perspective, limitations are inherited from the poor scalability of GPs (Rasmussen and Williams, 2006), for which inference becomes impractical when the number of runs of the code and the number of real observations are collectively beyond a few thousands. In addition, the use of Markov chain Monte Carlo (MCMC) (Neal, 1993) techniques to carry out inference for GP models can be painfully slow without careful tuning and clever parameterizations (Filippone et al., 2013; Filippone and Girolami, 2014).

This work aims to tackle these issues by proposing the use of recent developments in the GP and DGP literature and variational inference, (i) to extend the modeling capabilities of GPs in emulation using DGPs; (ii) to extend the original framework in Kennedy and O’Hagan (2001), by casting the model as a special case of a DGP; (iii) to adapt techniques based on random feature expansions and stochastic variational inference, building on the work by Cutajar et al. (2017), to obtain a scalable framework for Bayesian calibration of computer models. Thanks to this formulation, it is possible
We validate our proposal, which we name \textit{dgp-cal}, on a variety of calibration problems, comparing with alternatives from the state-of-the-art. We demonstrate the flexibility and the scalability of \textit{dgp-cal}, as well as the ability to capture the uncertainty in model parameters and model discrepancy. We conclude the paper by showing how we can carry out adaptive experimental design, and by discussing the identifiability properties of the proposed calibration model. The code to replicate all the results in the paper is available at the following url:

\url{https://github.com/SebastienMarmin/variational-calibration}.

2 Background

In this section, we introduce the problem of calibration of computer models and we present the \textit{koh} model. We then introduce Gaussian processes (GPs), which are the main modeling ingredients in the \textit{koh} model. Motivated by the difficulties associated with carrying out inference with GPs, we present random feature expansions as a way to reduce complexity and being able to exploit recent advances in approximate inference. In particular, we focus on variational inference (vi) techniques that are able to operate on mini-batches of data and that can be easily implemented in developing environments featuring automatic differentiation. We conclude this section by showing how we can increase flexibility of GPs by composing processes, obtaining DGP, for which we can extend the use of random feature expansions and vi. This background material gives us all the elements to present our proposed calibration model in Section 3.

2.1 Bayesian Calibration

Consider prediction and uncertainty quantification for a phenomenon approximated by a computer model, which is expensive to evaluate. Throughout the paper, we will assume that the output of the computer model is univariate, but we will discuss ways in which we can deal with multiple responses. We denote observations from the real phenomenon of interest by \( y \in \mathbb{R} \), and we assume that we have \( n \) of these available \( y = [y_1, \ldots, y_n]^T \) for a number of inputs \( X = [x_1, \ldots, x_n]^T \), with \( x_i \in D_1 \subset \mathbb{R}^{d_1} \). For example, in climate modeling, \( y \) could correspond to temperature measurements at \( n \) locations identified by latitude and longitude (in this case, \( D_1 = [-90, 90] \times [-180, 180] \)). The computer model simulating the real phenomenon requires the so-called calibration parameters \( \theta \in D_2 \subset \mathbb{R}^{d_2} \), as well as input variables \( x \in D_1 \). Calibration parameters may have a physical meaning (e.g., exchange rates determining the carbon cycle) and inference over these is a central goal of Bayesian calibration.

Beside the observations \( y \) associated with \( X \), the computer model is run at (possibly different) inputs \( X^* = [x_1^*, \ldots, x_N^*]^T \) and calibration parameters \( T = [t_1, \ldots, t_N]^T \), yielding a collection of outputs \( z = [z_1, \ldots, z_N]^T \). Note that we denote by \( T \) the collection of \( N \) parameter configurations at which the computer model is run, while \( \theta \)
denotes the true (unknown) parameter we are interested in inferring. Generally $N$ is larger than $n$ as running the computer model is easier compared to obtaining real world observations (albeit computationally expensive).

Following the definition of the KOH model in Kennedy and O’Hagan (2001), we assume that $y$ and $z$ are drawn from $p(y_i | f_i)$ and $p(z_j | \eta_j^*)$, which determine the likelihood functions. The vectors $f = [f_1, \ldots, f_n]^\top$ and $\eta^* = [\eta_1^*, \ldots, \eta_N^*]^\top$ result from mapping $(x_i, \theta)_i$ and $(x_j^*, t_j)_j$ through random functions $f$ and $\eta$, respectively. The link between the computer model with latent representation $\eta$, and the real phenomenon with latent representation $f$, is modeled by

$$f(x, t) = \eta(x, t) + \delta(x),$$

where $\delta(x)$ represents the discrepancy between the computer model and the real process. Figure 1 illustrates the KOH calibration model.

![KOH calibration model](image)

In their Bayesian formulation, Kennedy and O’Hagan (2001) assume $\eta(x, t)$ and $\delta(x)$ to be independent Gaussian processes (GPs); in other words, the KOH model assumes a given prior distribution over these functions, which takes the form of a GP. In addition, they assume a prior over $\theta$, and they aim to characterize the posterior distribution over $\theta$ given the observations of the real process and runs of the computer model. In order to keep the notation uncluttered, we denote by $\psi$ the set of GP parameters for $\eta(x, t)$ and $\delta(x)$, and we denote by $U$ the collection of all input locations $X, X^*, T$. The marginal likelihood of the KOH model is

$$p(y, z | U, \psi) = \int p(y | \eta_0 + \delta)p(z | \eta^*)p(\delta | X, \psi)p(\eta_0, \eta^* | \theta, U, \psi)p(\theta)d\eta^* d\delta d\eta_0 d\theta,$$

with vectors $\eta^* = [\eta(x_1^*, t_1), \ldots, \eta(x_N^*, t_N)]^\top$, $\eta_0 = [\eta(x_1, \theta), \ldots, \eta(x_n, \theta)]^\top$ and $\delta = [\delta(x_1), \ldots, \delta(x_n)]^\top$. The high dimensionality and the nontrivial dependence of this integrand with respect to the parameters of interest $\theta$, makes their inference intractable,
S. Marmin and M. Filippone

Thus requiring approximations. Similar considerations can be made for the predictive distribution, that is the one made by the KOH model at new input values. In the original work by Kennedy and O’Hagan (2001), inference is carried out using MCMC, which offers guarantees of convergence to the true posterior distribution over model parameters, but it can be extremely slow and impractical when the number of field observations and computer simulations is large. This is due to the poor scalability properties of GPs, which adds up to the need to repeatedly solve these within MCMC.

It is worth mentioning the identifiability issues associated with the KOH model, brought up in the discussion of the paper of Kennedy and O’Hagan (2001). Such issues arise due to the over-parameterization of the model, whereby it is possible to confound the effects of calibration parameters and model discrepancy. In particular, the KOH calibration model yields a joint posterior distribution over $\theta$ and $\delta(x)$, but as the number of observations increases, the KOH model concentrates this posterior over the manifold

$$\mathcal{M} = \{(t, \delta(x)) | \eta(x, t) + \delta(x) = f(x, t) \text{ with } x \in X \cup X^*\}.$$ 

Brynjarsdóttir and O’Hagan (2014) nicely illustrate this problem as they study how removing the discrepancy would result in a model where the estimation of model parameters $\theta$ is biased. The argument is that if $\eta(x, \theta)$ does not model the real physical process exactly and there is no discrepancy, the estimate of $\theta$ would be based on a misspecified model. Increasing the number of observations would not cure this fundamental issue with model misspecification. The conclusion is that discrepancy in the KOH model is necessary to hope for a sound inference over calibration parameters $\theta$ of the computer model, and imposing good priors on $\theta$ and $\delta(x)$ becomes of fundamental importance to mitigate the lack of identifiability. Alternatively, one can improve identifiability when multiple responses are available, and these are mutually dependent on the same set of calibration parameters $\theta$ (Arendt et al., 2012). In the literature, there are works addressing the issue of identifiability of the KOH models with alternative formulations, such as loss minimization (Tuò and Wu, 2016) or frequentist formulations (Wong et al., 2017). Within this line of works, assuming that the optimal $\theta$ is the optimizer of a loss function, Plumlee (2017) showed that the prior over $\delta(x)$ should be orthogonal to the gradient of the computer model. We remark that the issue of identifiability affects in a similar way the model that we propose here, and we dedicate part of the discussion to comment on how we can adopt current strategies from the literature to deal with this.

2.2 Gaussian Process and Random Features Expansions

A Gaussian process (GP) is a set of random variables such that any subset of these is jointly distributed as a Gaussian (Rasmussen and Williams, 2006). This definition makes them suitable for assigning priors over functions. Imposing a GP prior over a function $g_{\text{exact}}(u), u \in \mathcal{D} \subset \mathbb{R}^d$ means assigning a prior over the realizations of the function $[g_{\text{exact}}(u_1), \ldots, g_{\text{exact}}(u_l)]^\top$ at a set of $l$ inputs $u_1, \ldots, u_l$, such that this is multivariate Gaussian; this is because of the properties of marginals of multivariate Gaussian distributions. What needs to be specified is a mean function and a covariance function $c(u_i, u_j)$, which determines how realizations of the function at different inputs
covary and therefore the properties of the functions that can be drawn from the GP. For simplicity, we assume a constant zero mean, but adding a parametric mean function is straightforward. Inference in models involving GPs quickly becomes intractable when \(l\) grows beyond a few thousands. The reason is that sampling from GPs and posterior inference requires algebraic operations with the covariance matrix obtained by evaluating the covariance function among all possible pairs of inputs. These operations usually involve \(O(l^3)\) operations, and require storing \(O(l^2)\) entries for the covariance matrix. In this work, we bypass these limitations by making a model approximation which lowers both complexities to \(O(l)\), as we discuss shortly.

Note that in this short presentation of GPs we assume that the function is univariate, that is \(g_{\text{exact}}(u) \in \mathbb{R}\). Extending this to multivariate functions \(g_{\text{exact}}(u) \in \mathbb{R}^o\) is rather straightforward. What needs to be specified is a richer covariance structure, which is able to characterize the covariance between \((g_{\text{exact}})_r(u_i)\) and \((g_{\text{exact}})_s(u_j)\); see, e.g., Álvarez and Lawrence (2011) for an extensive treatment of these scenarios. When we assume a zero covariance across functions, we are effectively modeling each \(g_r(u)\) as an independent GP. We are free to parameterize each GP separately, or to use a common covariance \(c(u_i, u_j)\), so that covariance parameters are shared across the \(o\) functions. In this paper we prefer this latter approach to avoid introducing too many parameters, although we can easily incorporate more advanced modeling assumptions in our implementation.

For a large class of covariance functions, it is possible to show that draws from the GP prior are a linear combination of a possibly infinite number of basis functions with Gaussian-distributed weights (Neal, 1996; Rasmussen and Williams, 2006). This can be formulated for \(u \in \mathcal{D}\) as an infinite sum

\[
g_{\text{exact}}(u) = \phi_\infty(u) \mathbf{w}_\infty, \quad (2.2)
\]

with \(w_\infty\) infinite dimensional random vector with i.i.d. standard normal components, and \(\phi_\infty\) the evaluation of an infinite set of basis functions at \(u\). The exact covariance of \(g_{\text{exact}}\) is readily obtained as \(\forall u, u' \in \mathcal{D}\)

\[
c(u, u') = \mathbb{E}[\phi_\infty(u) \mathbf{w}_\infty \mathbf{w}_\infty^\top \phi_\infty(u')] = \phi_\infty(u) \mathbf{w}_\infty \phi_\infty(u'). \quad (2.3)
\]

The infinite representation induced by the covariance function suggests a way to approximate GPs by means of a finite dimensional truncation of \(\phi_\infty(u)\), which we denote by \(\phi(u) \in \mathbb{R}^p\), so that

\[
c(u, u') = \phi_\infty(u) \mathbf{w}_\infty \phi_\infty(u') \approx \phi(u) \mathbf{w}_\infty \phi(u'). \quad (2.4)
\]

The function \(g_{\text{exact}}(u)\) is then approximated by

\[
g_{\text{exact}}(u) \approx g(u) = \phi(u) \mathbf{w}. \quad (2.5)
\]

When using GPs in modeling problems with \(l\) observations, the truncation has the advantage of avoiding the need to solve expensive algebraic operations with the covariance matrix. Instead, the truncation turns GPs into generalized linear models. In order to retain the probabilistic flavor of GPs, it is natural to treat these models in a Bayesian way,
and this requires algebraic operations with matrices of size $p \times p$ (cost $\mathcal{O}(p^3)$), while the complexity with respect to $l$ is linear.

Random feature expansions (Rahimi and Recht, 2008; Lázaro-Gredilla et al., 2010) offer an elegant framework to construct a finite $p$-dimensional representation $\phi(u)$, which are referred to as the random features. As a working example, throughout this paper we consider the Gaussian covariance (or kernel) function, also known as the squared exponential or radial basis covariance function:

$$c(u, u') = \sigma^2 \exp \left(-\frac{1}{2}(u - u')^\top A^{-1}(u - u')\right). \tag{2.6}$$

The symmetric positive definite matrix $A$ controls the scaling and mixing of the inputs, whereas $\sigma^2$ controls the marginal variance of the $\text{GP}$. When the covariance is shift-invariant, it is possible to express the covariance as the Fourier transform of a positive measure (Rahimi and Recht, 2008). Applying the Fourier transform to the Gaussian covariance, and defining $\iota = \sqrt{-1}$, we obtain:

$$c(u_i, u_j) = \sigma^2 \int p(\omega) \exp (\iota(u_i - u_j)^\top \omega) \, d\omega, \tag{2.7}$$

which immediately suggests that the distribution $p(\omega)$ is also Gaussian, and it has the form $p(\omega) = \mathcal{N}(0, A)$. By sampling from $p(\omega)$, we can approximate the integral in the Fourier formulation by means of Monte Carlo, thus obtaining a finite dimensional representation of the covariance:

$$c(u_i, u_j) \approx \frac{2\sigma^2}{N_{RF}} \sum_{r=1}^{N_{RF}/2} \exp \left(\iota(u_i)^\top \tilde{\omega}_r\right) \exp \left(-\iota(u_j)^\top \tilde{\omega}_r\right). \tag{2.8}$$

In this expression, we sampled $N_{RF}/2$ values of $\omega$, denoted by $\tilde{\omega}_r$, and exploited the property of shift invariance to split the complex exponential in two parts. While the basis functions are complex, by realizing that the left-hand side is a real number, with simple trigonometric manipulations of the right-hand side, it is possible to express the previous equation in an equivalent form as:

$$c(u_i, u_j) \approx \frac{2\sigma^2}{N_{RF}} \sum_{r=1}^{N_{RF}/2} \left[\sin (u_i^\top \tilde{\omega}_r), \cos (u_i^\top \tilde{\omega}_r)\right] \left[\sin (u_j^\top \tilde{\omega}_r), \cos (u_j^\top \tilde{\omega}_r)\right]^\top. \tag{2.9}$$

Therefore, introducing $\phi : \mathbb{R}^{N_{RF}} \to \mathbb{R}^{N_{RF}}$ as the element-wise application of sine (for the first $N_{RF}/2$ components) and cosine (for the last $N_{RF}/2$ components), the resulting basis functions are

$$\phi(u) = \sqrt{\frac{2\sigma^2}{N_{RF}}} \left[\sin (u^\top \Omega), \cos (u^\top \Omega)\right]^\top, \tag{2.10}$$

where $\Omega = [\omega_1, \ldots, \omega_{N_{RF}/2}]$, and the sin and cos functions are applied element-wise to their argument. The basis functions are also called random features, because they
are obtained by multiplying the inputs \( u_i \) with a random matrix \( \Omega \), followed by the application of a nonlinearity. Similar considerations can be made, for instance, for the Matérn covariance, where the \( \omega \)'s are sampled according to a multivariate Student-\( t \) distribution. See also Cho and Saul (2009) for an alternative derivation showing how the arc-cosine covariance of order one can be approximated in a similar fashion by sampling \( \omega \) from \( p(\omega) = N(0,A) \) and by employing a rectified linear unit nonlinearity (\( h(x) = x \) if \( x > 0 \) and \( h(x) = 0 \) otherwise), which is very popular in the literature of deep neural networks.

2.3 Deep Gaussian Processes and Random Feature Expansions

A Deep Gaussian Process (DGP) is defined as a composition of functions:

\[
g_{\text{deep}}(u) = g^{(L)}(g^{(L-1)}(\ldots (g^{(1)}(u)) \ldots)),
\]

where each function \( g^{(i)}(\cdot) \) is assigned a GP prior (Damianou and Lawrence, 2013; Neal, 1996). Again, for simplicity we focus our presentation on univariate GPs, but we will discuss ways in which we can deal with multivariate GPs shortly. The composition can be interpreted as a way to feed the output of \( g^{(i)} \) as input to another \( g^{(i+1)} \). In a parallel with deep neural networks, each GP can be thought of as a “layer”. The composition operation makes DGP priors substantially different from GPs; that is, the realizations of \( g^{(L)}(u) \) at \( U \) are no longer multivariate Gaussian in general. We refer the reader to Neal (1996); Duvenaud et al. (2014); Matthews et al. (2018) for some in-depth discussions and illustrations on the composition of GPs.

The DGP prior induced by the choice of GP priors over the functions in the composition can be used as a prior over functions in statistical models. After choosing an appropriate likelihood function, one is usually interested in optimizing all model parameters, which include the covariance parameters of the GPs at all layers, characterizing the posterior over \( g^{(L)}(u) \) at \( U \), and making predictions for any \( u^* \). For DGPs, these tasks are analytically intractable due to the nontrivial dependence introduced by the composition. Most of the literature on DGPs extends approximations and inference techniques developed for “shallow” GPs. For instance, Hensman and Lawrence (2014); Salimbeni and Deisenroth (2017) extend the use of Nyström-type approximations (also known as inducing points approximations) to DGPs and carry out inference using variational techniques, whereas Bui et al. (2016) employs expectation propagation.

This work focuses on random feature expansions for DGPs instead, which were proposed and studied in Gal and Ghahramani (2016); Cutajar et al. (2017). In this framework, each GP in the composition is approximated by means of random features, as shown in the previous section. With this approximation, each GP layer becomes a linear model with a given distribution over the weights. Denoting by \( a \) the input to layer \( (i) \), the GP at the \( i \)th layer approximated with random features implements the following operations:

\[
\phi^{(i)}(a) = \sqrt{\frac{2(\sigma^2)^{(i)}}{N_{RF}^{(i)}}} [\sin(\Omega(a)^T), \cos(\Omega(a)^T)]^T, \quad g^{(i)}(a) = (\phi^{(i)})^T w^{(i)}. \quad (2.12)
\]
With this approximation, each GP layer can be seen as a two-layer neural network. The first layer implements a multiplication by a random matrix \( \Omega^{(i)} \) and applies a non-linearity by means of trigonometric functions. The second layer implements a linear combination of the inputs. Therefore, composing these approximate GPs gives rise to a particular form of a Bayesian deep neural network. In traditional deep neural networks nonlinearities are applied at each layer and all the weights are optimized; in the approximate DGP viewed as a Bayesian deep neural network, nonlinearities are applied every other layer, and only the \( \mathbf{w}^{(i)} \) weights are inferred, whereas the \( \Omega^{(i)} \) are random. See Cutajar et al. (2017) for an in-depth discussion on the connection with Bayesian deep neural networks and other ways in which \( \Omega^{(i)} \) can be treated in order to improve performance.

The model approximation with random features bypasses the challenges of carrying out inference having to deal with the composition of GPs, but the composition of the resulting linear models is still intractable from a Bayesian perspective due to the nonlinearities introduced by the basis functions. In the next section, we present variational inference as a way to derive a tractable and scalable inference scheme for DGPs approximated with random features.

### 2.4 Stochastic Variational Inference

In this work, we make use of Variational Inference (VI) techniques to carry out inference over model parameters. We give a brief overview of VI here, and we will show how this is applied to the proposed calibration model in the next section. We consider a modeling problem for a set of \( l \) pairs of input/labels observations \( (\mathbf{u}_i, v_i) \), with \( \mathbf{u} \in \mathbb{R}^d \), and \( v \in \mathbb{R} \). Let \( U = [\mathbf{u}_1, \ldots, \mathbf{u}_l]^\top \) and \( v = [v_1, \ldots, v_l]^\top \). Imagine developing a statistical model with parameters \( \Theta \) and with likelihood function \( p(\mathbf{v}|\mathbf{U}, \Theta) \), and assume a prior \( p(\Theta) \).

VI is useful when the posterior over \( \Theta \), that is \( p(\Theta|\mathbf{v}, \mathbf{U}) \) is intractable. In VI, an approximation \( q(\Theta) \in \mathcal{Q} \) to the posterior is introduced, and the objective is to make it as close as possible to the actual posterior \( p(\Theta|\mathbf{v}, \mathbf{U}) \). The standard way to do so in VI is to set up the following optimization problem:

\[
\arg\min_{q(\Theta) \in \mathcal{Q}} \{ D_{\text{KL}} [q(\Theta) \| p(\Theta|\mathbf{v}, \mathbf{U})] \},
\]

where \( D_{\text{KL}} \) is the Kullback-Leibler divergence measuring how different the two distributions are. With simple manipulations, it is possible to show that an equivalent problem is the one of maximizing the following lower bound to the log-marginal likelihood with respect to \( q(\Theta) \) (see, e.g., Jordan et al. 1999; Graves 2011; Blei et al. 2017):

\[
\log p(\mathbf{v}|\mathbf{U}) \geq \mathbb{E}_{q(\Theta)}[\log(p(\mathbf{v}|\Theta, \mathbf{U}))] - D_{\text{KL}} [q(\Theta) \| p(\Theta)].
\]

In other words, a candidate \( q(\Theta) \) providing the highest lower bound also minimizes the divergence to the exact posterior.

With an expression for the lower bound of the marginal likelihood, we can now attempt to maximize it with respect to \( q(\Theta) \), which generally means to optimize it
DGPs for Calibration of Computer Models

w.r.t. its parameters. Therefore, the family of distributions, $Q$, for the candidate approximation $q(\Theta) \in Q$ needs to be chosen before inference. While the approximation is constrained by the choice of the family $Q$, the complexity of $Q$ impacts the complexity of the lower bound maximization. The trade-off between the inference speed and the quality of the approximation is an active research domain. For instance Rezende and Mohamed (2015) and Liu and Wang (2016) increase the expressiveness of the variational distribution while keeping the inference tractable, at the cost of increasing the number of parameters of $q(\Theta)$.

The lower bound contains two terms: the first is a model fitting term, whereas the second is a regularization term which penalizes approximations that deviate too much from the prior. This divergence term can be computed analytically when priors and approximate posteriors have particular forms. For example, when $p_1(x) = \mathcal{N}(\mu_1, \sigma_1^2)$ and $p_2(x) = \mathcal{N}(\mu_2, \sigma_2^2)$, the Kullback-Leibler divergence between the two has the form:

$$D_{KL}(p_1(x) \parallel p_2(x)) = \frac{1}{2} \left[ \log \left( \frac{\sigma_2^2}{\sigma_1^2} \right) - 1 + \frac{\sigma_1^2}{\sigma_2^2} + \frac{(\mu_1 - \mu_2)^2}{\sigma_2^2} \right]. \quad (2.15)$$

The first term in the lower bound (2.14) depends on $q(\Theta)$ through the expectation of the log-likelihood. This complication is usually bypassed by employing stochastic optimization using Monte Carlo:

$$\mathbb{E}_{q(\Theta)}[\log(p(\mathbf{v}|\Theta, U))] \approx \frac{1}{N_{MC}} \sum_{k=1}^{N_{MC}} \log \left( p(\mathbf{v}|\hat{\Theta}_{(k)}, U) \right), \quad (2.16)$$

with $\hat{\Theta}_{(k)}$ i.i.d. samples from $q(\Theta)$. The Monte Carlo approximation is unbiased, and so it is its derivative with respect to any of the parameters of $q(\Theta)$. This means that we can employ stochastic gradient optimization to adapt the parameters of $q(\Theta)$ to maximize the lower bound with guarantees to reach a local optimum of the objective (Robbins and Monro, 1951; Graves, 2011). The only precaution to take to make this viable, is to reparameterize the samples from $q(\Theta)$ using the so-called reparameterization trick (Kingma and Welling, 2014). In its simplest form, assuming a fully factorized Gaussian posterior over all parameter components $\theta_\ell \in \Theta$, the expression $(\hat{\theta}_\ell)_{(k)} = \mu_\ell + (\hat{\epsilon}_\ell)_{(k)} \sigma_\ell$ separates out the stochastic $((\hat{\epsilon}_\ell)_{(k)} \sim \mathcal{N}(0,1))$ and deterministic ($\mu_\ell$ and $\sigma_\ell$) components in the way samples from the approximate posterior are generated. In this way, the lower bound can be differentiated with respect to the variational parameters $\mu_\ell$ and $\sigma_\ell$ (with the $(\hat{\epsilon}_\ell)_{(k)}$ variables fixed), and it is therefore possible to perform gradient-based optimization (Graves, 2011; Kingma and Welling, 2014; Cutajar et al., 2017). The gradients are stochastic because the $(\hat{\epsilon}_\ell)_{(k)}$ variables are random, but the Monte Carlo estimate of the objective guarantees that the estimate of the gradient is unbiased, allowing for the use of stochastic gradient-based optimization (Robbins and Monro, 1951). This is the reason why this implementation of VI is also referred to as stochastic variational inference (SVI).

Finally, it is worth noting that it is possible to considerably reduce the variance of the stochastic gradients, thus increases convergence speed of the optimization by means of the so-called local reparameterization trick (Kingma et al., 2015). In this approach, instead of sampling $(\hat{\epsilon}_\ell)_{(k)}$, one samples from the distribution of the product of the inputs to a layer and samples from $q(\Theta)$; see Kingma et al. (2015) for more details.
Part of the huge success of deep learning is due to the exploitation of mini-batch-based optimization and automatic differentiation (Graves, 2011). The former enables scalability, as the model is updated by iteratively processing subsets of data. The latter, instead, allows one to tremendously simplify the implementation of complex models, as one has to implement the objective function and automatic differentiation takes care of computing its derivatives based on the graph of computations. Naively operating with mini-batches in GPs ignores the covariance among observations, which is crucial for effective GP modeling.

The proposed GP and DGP approximation and SVI allow us to exploit mini-batch-based optimization. Assuming that the likelihood factorizes across observations, \( p(v | \Theta, U) = \prod_j p(v_j | \Theta, U) \), the terms within the approximation of the fitting term in the lower bound can be estimated without bias by selecting \( I \), a set of \( m \) out of \( n \) indices (Graves, 2011)

\[
\mathbb{E}_{q(\Theta)}[\log (p(v | \Theta, U))] \approx \frac{1}{N_{MC}} \sum_{i=1}^{N_{MC}} \frac{1}{m} \sum_{\ell \in I} \log \left( p(u_\ell | \tilde{\Theta}_k, U) \right). \tag{2.17}
\]

This approximation introduces an extra level of stochasticity in the optimization (without introducing bias), but it allows one to scale the inference of these models to virtually any number of observations; previous work has reported results on DGPs for \( 10^7 \) observations with a single-machine implementation (Cutajar et al., 2017). Another important remark is that this approximation and inference approach for DGPs can be implemented relying exclusively on matrix-matrix and matrix-vector products, which can be accelerated by using GPU-type hardware.

3 DGPs for Calibration of Computer Models

In this section we present our contribution, which we refer to as DGP-CAL. We begin by observing that the KOH calibration model can be seen as a special case of a DGP, and this allows us to generalize the original formulation of the KOH model to more flexible ones. We then show how we can leverage the advances in approximation and inference for DGPs presented in the previous sections, namely random feature expansions and stochastic variational inference, in order to obtain a scalable framework for calibration, while retaining the flexibility offered by the use of DGPs. We conclude the section by discussing implementation details.

3.1 Generalization of the KOH Calibration Model as a DGP

The original formulation of the KOH calibration model involves the use of GPs to emulate the computer model and to model the additive discrepancy. As pointed out by Kennedy and O’Hagan (2001), additive discrepancy is somewhat specific and it can be generalized (see, e.g., Qian and Wu 2008). We propose to do so by assuming that the function underlying the real process is obtained by a warping function \( \gamma \) applied to the emulator:

\[
f(x, t) = \gamma (\eta(x, t), x). \tag{3.1}
\]
We retrieve the KOH formulation (2.1) when the warping applies the identity to \( \eta(\mathbf{x}, \mathbf{t}) \) and adds it to a GP on \( \mathbf{x} \). In other words, we can think of the function \( f(\mathbf{x}, \mathbf{t}) \) in the KOH model as a composition of two functions; the first is a GP that, given the inputs \( \mathbf{x} \) and \( \mathbf{t} \), yields \( \eta(\mathbf{x}, \mathbf{t}) \), whereas the second applies a linear combination of \( \eta \) and another GP \( \delta(\mathbf{x}) \) with fixed unit weights.

In the original formulation of the KOH model, the discrepancy between the computer code and the real process is modeled through the additive discrepancy term \( \delta(\mathbf{x}) \). In the proposed generalization of the KOH model, instead, we assume a GP prior over \( \gamma(\eta(\mathbf{x}, \mathbf{t}), \mathbf{x}) \), so that we apply a warping as a function of \( \mathbf{x} \). Similarly to the original KOH model, the analysis of the warping function \( \gamma(\eta(\mathbf{x}, \mathbf{t}), \mathbf{x}) \) allows one to reason about the discrepancy between the computer model and the real process.

A further possible extension, which we implement in our work, is to increase model flexibility by letting \( \eta(\mathbf{x}, \mathbf{t}) \) and \( \eta(\mathbf{x}, \mathbf{t}), \mathbf{x} \) to be modeled as DGP s instead of GPs. The deep extension is particularly useful when the emulator or the real process exhibit a space-dependent behavior that is difficult to model by designing appropriate covariance functions. DGPs offer a way to learn such nonstationarities from data, so this is particularly appealing in such challenging applications. We will give illustrations of DGPs and the generalized formulation in the experiments. Thanks to the possibility to employ random feature approximations of the DGPs in the generalized model, we can obtain a scalable framework for calibration as discussed next.

### 3.2 Model Approximation Using Random Features

In this section, we discuss how to employ the random feature approximation to make the KOH model and its generalization suitable for variational inference. We start from the KOH model, assuming that \( \eta(\mathbf{x}, \mathbf{t}) \) and \( \delta(\mathbf{x}) \) are modeled as GPs with the Gaussian covariance in (2.6); we denote their anisotropy matrices by \( A_\eta \) and \( A_\delta \) and their marginal variances \( \sigma^2_\eta \) and \( \sigma^2_\delta \), respectively. By applying the random feature expansion detailed in Section 2.2 we obtain

\[
\eta(\mathbf{x}, \mathbf{\theta}) = \phi_\eta(\mathbf{x}, \mathbf{\theta})^\top \mathbf{w}_\eta = \sigma_\eta \phi \left( \Omega_\eta \begin{bmatrix} \mathbf{x} \\ \mathbf{\theta} \end{bmatrix} \right)^\top \mathbf{w}_\eta, \tag{3.2}
\]

\[
\delta(\mathbf{x}) = \phi_\delta(\mathbf{x})^\top \mathbf{w}_\delta = \sigma_\delta \phi \left( \Omega_\delta \mathbf{x} \right)^\top \mathbf{w}_\eta. \tag{3.3}
\]

The feature maps \( \phi_\eta \) and \( \phi_\delta \) use the functions \( \phi : \mathbb{R}^{N_{\text{RF}}} \to \mathbb{R}^{N_{\text{RF}}} \) given in (2.10). The elements of \( \mathbf{w}_\eta \) and \( \mathbf{w}_\delta \), of size \( N_{\text{RF}} \), have i.i.d. standard normal priors, whereas the matrices \( \Omega_\eta, \Omega_\delta \) of size \( N_{\text{RF}} \times (d_1 + d_2) \), \( N_{\text{RF}} \times d_1 \), have i.i.d. normal rows, with covariance dependent on the positive definite matrices \( A_\eta \) and \( A_\delta \); in particular, the columns of \( \Omega_\eta \) and \( \Omega_\delta \) are i.i.d. \( \mathcal{N}(\mathbf{0}, A_\eta) \) and \( \mathcal{N}(\mathbf{0}, A_\delta) \), respectively. Figure 2 represents the model (using a neural network-like diagram) according to (2.1), (3.2), and (3.3).

It is straightforward to extend this formulation to model \( \eta(\mathbf{x}, \mathbf{t}) \) and \( \delta(\mathbf{x}) \) with DGPs instead of GPs, by applying the random feature expansion to each GP layer. Assuming that each DGP has \( L_\eta \) and \( L_\delta \) layers, in this case, \( \eta(\mathbf{x}, \mathbf{t}) \) and \( \delta(\mathbf{x}) \) are approximated by
Figure 2: Neural Network representation of the proposed approximation to the KOH model. (3.2) and (3.3) formulate random feature expansions for \( \eta(x, t) \) and \( \delta(x) \).

a Bayesian deep neural network, where each GP layer is approximated by:

\[
\eta(x, \theta) = \phi^{(L_\eta)}(a^{(L_\eta)})^\top w^{(L_\eta)} \quad \phi^{(L_\eta)}(a^{(L_\eta)}) = \phi(\Omega_\eta a^{L_\eta}),
\]

\[
a^{L_\eta} = \phi^{(L_{\eta-1})}(a^{(L_{\eta-1})})^\top w^{(L_{\eta-1})} \quad \phi^{(L_{\eta-1})}(a^{(L_{\eta-1})}) = \phi(\Omega_\eta a^{L_{\eta-1}}),
\]

\[
... = ...
\]

\[
a^{(2)} = \phi^{(1)}(x, \theta)^\top w^{(1)} \quad \phi^{(1)}(x, \theta) = \phi(\Omega_\eta^{(1)} \begin{bmatrix} x \\ \theta \end{bmatrix}) . \tag{3.4}
\]

Similarly, we can construct a random feature approximation for a DGP modeling \( \delta(x) \) in the KOH model, or apply the same construction for the warping function \( \gamma(\eta(x, t), x) \) in the generalized version of the KOH model. The advantage of the proposed approximation using random features is that it enables the use of stochastic variational inference presented in Section 3.3, as discussed next.

### 3.3 Inference of the Approximate DGP Calibration Model

Again, we focus on the formulation of the KOH model with the discrepancy term \( \delta(x) \) instead of the warping \( \gamma(\eta(x, t), x) \), but it is easy to follow the same derivation for this latter case. We make use of variational inference (VI), so we approximate the posterior distribution over all model parameters \( w_\eta, w_\delta, \) and \( \theta \) by introducing a variational posterior \( q(w_\eta, w_\delta, \theta) \), see Section 2.4. We assume \( q(w_\eta, w_\delta, \theta) \) to be Gaussian and
completely factorized across parameters, that is
\[
q(w_\eta, w_\delta, \theta) = \prod_{i=1}^{N_{RF}} \prod_{j=1}^{d_{\text{out}}} q(w_{\eta,ij}) \prod_{\ell=1}^{d_2} q(w_{\delta,ij}) \prod_{\ell=1}^{d_{\text{2}}} q(\theta_{\ell}),
\]
(3.5)

although the factorization assumption can be relaxed. With the assumption that \(q(w_\eta, w_\delta, \theta)\) factorizes across parameters, we deal with computations and storage which are linear in the number of model parameters. Relaxing the factorization means to assume that certain groups of parameters have nonzero covariance. For example, one could assume \(q(w_\eta, \cdot, j) = N(\mu_\eta,j, \Sigma_\eta,j)\) and parameterize \(\Sigma_\eta,j = L_\eta,j L_\eta,j^T\) to preserve positive-definiteness by optimizing \(L_\eta,j\). It is easy to verify that this choice requires \(O(N_{RF}^2)\) storage and \(O(N_{RF}^3)\) computations.

Following the principles of VI, we need to derive a lower bound to the marginal likelihood of the model and maximize it with respect to the distribution \(q(w_\eta, w_\delta, \theta)\). In practice, this problem turns into the optimization of the lower bound with respect to the parameters that govern \(q(w_\eta, w_\delta, \theta)\). Taking the lower bound (2.14) and its unbiased Monte Carlo and mini-batch-based approximation (2.17), we now adapt this to our calibration model. In this adaptation we realize that there are two types of input points, namely observations \(X\) and computer runs \([X^*, T]\); note that the shapes of \(X, X^*, T\) do not allow a concatenation of these three matrices in a single matrix. One possibility is to apply mini-batch on the union of the data sets. However we do not recommend this procedure without ensuring that each category (“Observations” versus “Runs”) gets sufficiently represented. For example a uniform sampling blind to the category of the input points with \(n < N\), would make the number of sampled observations vary a lot from one iteration to the other and may sometimes sample none of them. A workaround is to draw two index sets, \(I \subset \{1, \ldots, n\}\) and \(J \subset \{1, \ldots, N\}\) of sizes \(m\) and \(M\). The fitting term of the lower bound can now be approximated as

\[
\mathcal{E} := \mathbb{E}_{q(w_\eta, w_\delta, \theta)} \left[ \log(p(y, z|w_\eta, w_\delta, \theta, U)) \right]
\approx \frac{1}{N_{MC}} \sum_{k=1}^{N_{MC}} \left[ \frac{1}{m} \sum_{i \in I} \log \left( y_i | \psi, X, (\tilde{w}_\eta)_{(k)}, (\tilde{w}_\delta)_{(k)}, \tilde{\theta}_{(k)} \right) + \frac{N}{M} \sum_{j \in J} \log \left( z_j | \psi, X^*, T, (\tilde{w}_\eta)_{(k)} \right) \right],
\]
(3.6)

with \((\tilde{w}_\eta)_{(k)}, (\tilde{w}_\delta)_{(k)}, \tilde{\theta}_{(k)}\) i.i.d. samples from \(q(w_\eta, w_\delta, \theta)\). The regularization term can be easily calculated when both priors and posteriors are Gaussian using (2.15).

### 3.4 Implementation Details

Considering the large number of parameters to optimize, the optimization procedure is divided into stages. We first focus on the computer model response; all parameters are fixed except the ones influencing the prediction of \(z\), i.e. \(\sigma_z\), the means and variances of the components of \(w_\eta\), and the GP/DGP parameters of \(\eta(x, t)\). In the second stage
all others parameters are freed for inferring $y$ and $\theta$ jointly, i.e. adding the weights and hyperparameters of $\delta(x)$ or $\gamma(\eta(x, t), x)$. Within each stage, we first optimize the means and variances of $\theta$, and then all parameters jointly with a smaller learning rate.

For initializing the weights of the DGP (or GP) $\eta(x, t)$, we use the methodology proposed by Rossi et al. (2019), which is a scalable, layer-wise initialization strategy based on Bayesian linear regression. The idea is to perform a series of regression tasks mapping each layer to the labels, so that sensible initial parameters for the variational distribution can be obtained. Starting from the first layer, we perform Bayesian linear regression from the inputs $X^*$ and $T$ to the labels $z$, so that we can estimate the posterior over the parameters at the first layer. We then freeze this distribution and compute the output of the first layer with the given input data $X^*$ and $T$. This result is then used as the input to the second layer, for which the corresponding linear model is estimated by performing regression to the labels $z$. We then proceed iteratively up to the last layer. Intuitively, this initialization promotes configurations where the first layers are already capable of obtaining sensible regression result, while the layers closer to the labels serve as refinements.

We define the routine $\mu, \Sigma \leftarrow \text{LM}(X_{LM}, y_{LM})$, which performs Bayesian linear regression on a set of input-output pairs $X_{LM}, y_{LM}$. The routine returns the mean and the covariance of the posterior distribution over the weights $w$ in the instrumental regression $y_{LM} := z = X_{LM}w + \sigma^2_\eta \varepsilon$, where $\sigma^2_\eta$ is the variance of the likelihood function, and the prior for the components of $w$, $\varepsilon$ is i.i.d. $\mathcal{N}(0, 1)$. With these definitions, the initialization is reported in Algorithm 1. The posterior distribution in Bayesian linear regression has full covariance in general, whereas the assumed posterior is fully factorized. In this case, we match the optimal factorized Gaussian distribution to the actual posterior using the Kullback-Leibler divergence, which explains the assignment to $q(W^{(f)}_{0:j})$ in the last line of Algorithm 1. The procedure can easily exploit mini-batching, as reported in Algorithm 1, and it can operate with stochastic optimization, thus making it suitable for large-scale problems. We refer the reader to Rossi et al. (2019) for more details.

4 Experiments

In this section we validate DGP-CAL on a number of calibration problems. In each experiment, we specify whether DGP-CAL uses the additive structure in (2.1), as in the KOH formulation, or the general one in (3.1); we also specify when the model is tested with DGPs instead of GPs (the default). The experiments have the following setup. The likelihoods $p(y_i|f_i)$ and $p(z_j|\eta_0^*)$ are Gaussian with variances $\sigma^2_y$ and $\sigma^2_z$ treated as hyperparameters within $\psi$. All covariance functions of $\eta(x, t)$ and $\delta(x)$ are Gaussian, except for the comparative experiment in Section 4.2 where a Matérn kernel is used. The variational posteriors $q(W^f_\eta)q(W_\delta)q(\theta)$ and the prior $p(\theta)$ are Gaussian.

Competing Methods A large literature is devoted to the practicalities of numerically challenging applications. Gramacy et al. (2015) use local approximate GP modeling and calibrate parameters by solving a derivative-free maximization of a likelihood term. Pratola and Higdon (2016) handle large problems using a Bayesian sum-of-trees regression
DGPs for Calibration of Computer Models

Input:
- DGP \( \eta = g^{(L)}(g^{(L-1)}(\ldots (g^{(1)}(x, t)))) \) with variational distribution of the weights \( w_\eta = \{W^{(1)}_\eta, \ldots, W^{(L)}_\eta\} \) set to the prior \( \mathcal{N}(0, 1) \) i.i.d. for all components.
- Computer runs \( X^*, T \) with outputs \( z \) organized in mini-batches of size \( M \).

Result: A distribution \( q(w_\eta) \) to start the lower bound maximization with.

for each layer index \( \ell \) do
  for each output component \( i \) of \( g^{(\ell)} \) do
    • \( X^*_b, T_b, z_b \leftarrow \) next data batch;
    • Propagate \( X^*_b, T_b \) through a sample path of \( \eta \) and save \( X_{LM} \) as:
      \( X_{LM} \leftarrow [g^{(\ell-1)}(\ldots (g(x^*_b, t, j)))]_{j=1, \ldots, M} \)
      (\( X_{LM} \) is the output of the previous layer \( g^{(\ell-1)} \), uses \( [X_b^*, T_b] \) for \( \ell = 1 \));
    • \( y_{LM} \leftarrow z_b \);
    • \( \mu, \Sigma \leftarrow \text{LM}(X_{LM}, y_{LM}) \);
    • Initialize the variational distribution of the \( i \)th column of \( W^{(\ell)}_\eta \)
      \( q(W^{(\ell)}_{\eta,i}) \leftarrow \mathcal{N}(\mu, \text{diag}(\frac{1}{\Sigma_{ii}}, \ldots, \frac{1}{\Sigma_{MM}})) \);
  end
end

Algorithm 1: Initialization for the approximate posterior distribution over \( \eta \).

4.1 Illustrative Example

We illustrate DGP-CAL on a calibration problem with one variable and one calibration input. As a first test, the prior and hyperparameters used to generate the data set are assumed to be known, with \( \theta \sim \mathcal{N}(0, 1), \sigma_\eta = 1, A_\eta = \frac{1}{2} I, \sigma_\delta = \frac{2}{10}, A_\delta = \frac{1}{20} \). We choose locations for \( N = 7 \) computer runs and \( n = 4 \) observations from the real process in a space filling manner in \([0, 1] \times [-\frac{5}{2}, \frac{5}{2}]\). The output vector \( z \) of the computer model at \( (x^*_i)_{i=1, \ldots, N} \) is sampled from its prior distribution. In order to determine the real observations \( y \), we first sample \( p(\theta) \) to get \( \theta_{\text{true}} \) and then a sample path of \( \delta(x) \). The GP priors of \( \eta(x, t) \) and \( \delta(x) \) are approximated with \( N_{RF} = 50 \) random features through (3.2) and (3.3), and the observations are computed using (2.1). The results of DGP-CAL are displayed in Figure 3. In the first and the third panels, we see that the posterior of \( \theta \) obtained analytically by integrating out \( w_\eta \) and \( w_\delta \) has its mass
concentrated around the true value $\approx 0.8$, where there is a (color) match between $z$ (the dots) and $y$ (the lines). The variational posterior (blue line) offers a reasonable approximation of the true posterior.

Figure 3: **Top-left:** the prior (black), the analytical posterior (green) and the variational posterior (blue) distributions of $\theta$ and the actual value used to generate $y$ (red). **Top-right:** the true response $\eta$ used to generate the data set of this example and the locations of the computer runs (dots). **Bottom-left:** shows $y$ as horizontal lines in $D_1 \times D_2$. The colors of the lines correspond to the values $y_i$. The dots represent the computer runs $z$. The grey levels represent the posterior distribution of $\theta$ (also displayed in the top-left panel). **Bottom-right:** the posterior mean of $\eta$.

4.2 Model Calibration in Cell Biology

We apply dgp-cal to a biological application, which has been previously studied in Plumlee (2017) and Xie and Xu (2018). The output is the normalized current through ion channels of cardiac cells needed to maintain the membrane potential at $-35\, \text{mV}$. The input variable $x$ is the logarithm of the experiment time rescaled to $D_1 = [0,1]$. The calibration inputs $\theta \in D_2 = [0,10]^3$ control a mathematical model $\eta_{\text{cell}}(\cdot, \theta)$ of the phenomenon proposed by Clancy and Rudy (1999). Here it is considered to be an expensive black box with $N = 300$ runs available, whereas the number of observations is $n = 19$. The runs are located in a space filling manner in $D_1 \times D_2$ (Latin hypercube sampling optimized with maximin distance criterion).

We compare dgp-cal with additive and general discrepancy against four competitors. The method “$L_2$” is a simple minimization over $\theta$ of the $L_2$ residual error $||y - \hat{\eta}_{\text{cell}}(X, \theta)||$, where $\hat{\eta}_{\text{cell}}$ is a surrogate of $\eta_{\text{cell}}$ given $X^*$, $T$ and $Z$. Its minimization takes 30 seconds and the residual error is 1.31. This method is good for predicting observations from the real process, but it provides no quantification of uncertainty.

In Table 1, we report the mean squared error (MSE), $E_q(\theta)(||y - \eta_{\text{cell}}(X, \theta)||^2)$, where $q$ represents the estimated posterior density of $\theta$. All tuning parameters of the codes
are left to default values, and all methods are run on the same machine to ensure some fairness in reporting running times (laptop with $4 \times 2.50$ GHz cores). We see that the MSE values obtained by the methods PROJECTED, DGP-CAL and ROBUST are significantly lower than the MSE of KOH. The proposed DGP-CAL is the fastest. The version of DGP-CAL with general discrepancy performs slightly better, thanks to the relaxation of the hypothesis of purely additive discrepancy.

Table 1: Comparison of errors on the Cell Biology problem.

<table>
<thead>
<tr>
<th>Method</th>
<th>Time (s)</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROJECTED</td>
<td>3792</td>
<td>2.52</td>
</tr>
<tr>
<td>DGP-CAL ADDITIVE</td>
<td>79</td>
<td>1.83</td>
</tr>
<tr>
<td>DGP-CAL GENERAL</td>
<td>93</td>
<td>1.55</td>
</tr>
<tr>
<td>KOH</td>
<td>3245</td>
<td>5.10</td>
</tr>
<tr>
<td>ROBUST</td>
<td>361</td>
<td>1.99</td>
</tr>
</tbody>
</table>

The posterior distributions over $\theta$ obtained by the calibration methods we tested are reported in Figure 4. All methods yield a distribution concentrated around the $L_2$ minimizer (the red dot). However, the distributions are clearly not similar to each other (except for ROBUST and DGP-CAL). This could be explained by differences in the model formulations. Although we ensured that the covariance and mean functions are the same for all competing methods (Matérn with smoothness 5/2 and constant mean), there are several differences that cannot be matched. For instance, ROBUST has an additional step in the hierarchy of priors concerning the $L_2$ norm of the discrepancy. Moreover, the definition of the calibration parameters $\theta$ itself differs among methods. In PROJECTED, $\theta$ is a minimizer of a given stochastic process, while other methods follow the KOH definition. Also, ROBUST performs a fully Bayesian inference including hyperparameters, while in the other methods, including ours, they are optimized. It would be straightforward to allow for a Bayesian treatment of the hyperparameters in DGP-CAL, but we leave this for future work.

To visualize the results of the calibration process, in Figure 5 we overlay the observations from the real process with the responses of the computer model $\eta_{\text{cell}}(\cdot, \theta)$ when $\theta$ is sampled from its posterior distribution. All the probabilistic methods present a good fit while allowing for quantification of uncertainty in the predictions, with larger uncertainty for models that account for the uncertainty in the hyperparameters.

We see how the computer model output $\eta$ is warped by $\gamma$ in DGP-CAL with general discrepancy (3.1). In Figure 6 we display the expected derivative of the warping with respect to the computer model output, i.e., $E \left[ \frac{\partial \gamma(\cdot, x)}{\partial \eta} \right]$, for three values of $x$. As the estimated values oscillate around one for every $x \in D_1$, this model confirms that an additive discrepancy is a sensible assumption. When the estimated $\gamma(\cdot, x)$ is exactly the identity, the general discrepancy boils down to an additive one. This figure also shows how the model with general discrepancy can adapt to data sets with space-dependent behavior. Indeed in this test case the values of $\eta$ have a very different distribution according to $x$. If $x$ is around 0.2, the distribution of the computer runs is very asymmetric, with a
heavy tail for high values, and very light tail for low values (see the grey dots in Figure 5). On the other hand for higher values of \( x \), say higher than 0.6, the distribution of \( z \) is more symmetric, and looks closer to a Gaussian distribution. This corresponds to the warping observed in Figure 6, where \( \eta \) gets its output warped and concentrated asymmetrically toward lower values for \( x = 0.2 \), while for \( x = 0.6 \) or 1, its Gaussian output is almost left untouched.

4.3 Model Calibration for Complex Response

We deal now with a case-study with locally nonsmooth/nonstationary response of the computer model, for which a stationary \( \text{gp} \) is generally inadequate. The computer model is a simulator of the effects of underground nuclear tests on radionuclide diffusion into aquifers at the Yucca Flats in the United States (Fenelon, 2005). We take the same data set as generated by a script in the supplementary material of Pratola and Higdon (2016), which is available online, with \( d_1 = 2 \), \( d_2 = 6 \), \( n = 10 \) and \( N = 17600 \).

In Pratola and Higdon (2016), the size of the dataset as well as nonstationary modeling is handled with a sum-of-trees regression. We carry out calibration using DGP-CAL with a two-layer DGP emulator for the computer model to showcase the ability of a more complex emulator to capture the nonstationarity that characterizes this problem. We therefore compare DGP-CAL with a shallow \( \text{gp} \) emulator. Details about the initialization can be found in Table 2. Furthermore, we compare against the modularized method with Local Approximate GPs (LAGP) of Gramacy et al. (2015).

In Figure 7, we display the posterior over the function \( f(\cdot, \theta) \) modeling the real observations. We observe that only the deep variational calibration and the sum-of-trees approach manage to reproduce the nonstationary nature of the data set by capturing the spike characterizing one observation.
DGPs for Calibration of Computer Models

Figure 5: Samples of $\eta_{\text{cell}}(\cdot, \theta)$, with $\theta$ drawn from its posterior. The grey dots represent the computer runs and the white dots the real observations.

Figure 6: Derivative $\frac{\partial \gamma(\cdot,x)}{\partial \eta}$ for three values of $x$.

4.4 Data Set Size Scalability

We now showcase the scalability of DGP-CAL to a large calibration problem in 8 dimensions with one million computer runs, and 100,000 real observations. We use the borehole function $\eta_{bh}(x, t)$, which is a widely used function in the literature of computer experiments (see, e.g., Gramacy et al. 2015). For all $x \in [0, 1]^5$ and $t \in [0, 1]^3$, we have

$$
\eta_{bh}(x, t) = \frac{2\pi T_u(H_u - H_l)}{\ln(r/r_w) \left( 1 + \frac{2LT_u}{\ln(r/r_w) r_w K_w + \frac{T_u}{H_l}} \right)}, \quad \delta_{bh}(x) = \frac{2(10x_1^2 + 4x_2^2)}{50x_1x_2 + 10}, \quad (4.1)
$$

with $T_u = x_1(115600 - 63070) + 63070$, $H_u = x_2(1110 - 990) + 990$, $H_l = x_3(820 - 700) + 700$, $L = x_4(1680 - 1120) + 1120$, $K_w = x_5(12045 - 9855) + 9855$, $r_w = t_1(0.15 - 0.05) + 0.05$, $r = t_2(50000 - 100) + 100$, $T_l = t_3(116 - 63.1) + 63.1$. 
We are interested in retrieving a randomly chosen true value $\theta = [0.089, 0.308, 0.372]^T$. The locations $X$, $X^*$ and $T$ are generated with Latin hypercube sampling. To generate $y_i$, a white Gaussian noise $\varepsilon$ of standard deviation $\sigma_{bh} = 5 \times 10^{-3}$ is added: $y_i = \eta(x_i, \theta) + \delta(x_i) + \varepsilon_i$.

We build a shallow DGP-CAL model with additive discrepancy as described by (2.1), (3.2), and (3.3). Indeed, as the response surface of the borehole functions is smooth, we consider “shallow” GPs for $\eta(x, \theta)$ and $\delta(x)$, with Gaussian covariance approximated by 100 random features.

Concerning the sum-of-trees calibration, a sensible budget would be to set 2000 posterior samples plus 10000 for burn-in, with 1000 tree cut-points. However, this corresponds to one month of computation on our computers, so we divided the sampling budget by 5, and set 100 cut-points, keeping all other default parameters untouched.

We did not compare with the modularized calibration using LAGP, as the current implementation in R does not support large amount of real observations. This does not question the relevance of the method, which could be modified by using a scalable GP for the discrepancy.
Table 3: Results of calibration on a large data set.

<table>
<thead>
<tr>
<th>Calibration</th>
<th>Time (h)</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>None (unif. samples on $D_2$)</td>
<td>0</td>
<td>0.32</td>
</tr>
<tr>
<td>dgp-cal</td>
<td>1.4</td>
<td>0.03</td>
</tr>
<tr>
<td>Sum-of-trees</td>
<td>132.4</td>
<td>0.14</td>
</tr>
</tbody>
</table>

We evaluate the performance by comparing the posterior over $\theta$ with the truth (Figure 8) and by evaluating the MSE error between the computer model and observation from the real process $E_{q(\theta)}(||y - \eta_{bh}(X, \theta)||^2)$ (Table 3). DGP-CAL provides the best performance both in retrieving $\theta$ and MSE, and it is the fastest by far.

Figure 8: Posterior distribution of $\theta$ on a large data set (black: truth, blue: dgp-cal, green: sum-of-trees).

5 Discussion

The KOH model and inference in Kennedy and O’Hagan (2001) offers a classical framework to tackle calibration problems where quantification of uncertainty is of primary interest. In this paper we proposed DGP-CAL, which offers a number of improvements over the KOH calibration. From the modeling perspective, we cast the KOH calibration model as a special case of a more general DGP model, where the latent process modeling the real observations is a warped version of the emulator of the computer model; we showed that this general calibration model retains the possibility to reason about uncertainty in the discrepancy between the computer model and the real process.

Furthermore, the proposed approximation of GPs and DGPS with random features and approximate inference through variational techniques give DGP-CAL a number of advantages, such as simplicity of implementation in development environments featuring automatic differentiation and the possibility to exploit GPU-type hardware. The experiments showed that the approximations introduced to recover tractability do not affect the ability of DGP-CAL to effectively calibrate parameters of complex computer models, while enjoying scalability to large number of observations and/or computer runs, demonstrating that DGP-CAL is a powerful alternative to the state-of-the-art. We are currently investigating the application of DGP-CAL to other large-scale calibration problems in
environmental sciences, where the KOH model and related calibration methodologies are usually not the preferred choice due to limited scalability.

We conclude the paper by discussing two important aspects of this work, namely adaptive experimental design and identifiability.

**Adaptive Experimental Design** Building up the design sequentially may allow smaller training sets, limiting the evaluation of the simulator code (see e.g. Sauer et al. 2021). It is possible to extend DGP-CAL to handle cases where the uncertainty in the model can be used to guide the incremental design of the experiment. For simplicity, we assume $\eta(x, t)$ in the KOH model to be modeled as a GP. The goal is to improve calibration by sequentially optimizing the locations of the inputs of the computer model ($X^*$ and $T$). More precisely, instead of determining the $N$ locations from the outset, $N_{\text{add}}$ points are added at each of $n_{\text{it}}$ iterations to an initial design of $N_0 < N$ experiments. At each iteration, the model is inferred on the data integrating the new batch of $N_{\text{new}}$ inputs and outputs. The batch of points $[X^*_\text{ca}, T_{\text{ca}}]$ with corresponding labels $z_{\text{ca}}$ added to the design is determined as a solution to the optimization of a criterion. We can specify a criterion such that the evaluation of $\eta$ at the best candidate points maximally reduces the variance $\theta$. Many sampling criteria can be imagined and tested; for example, one can choose the sum of the partial derivatives of the lower bound with respect to the (logarithm of the) variance of the components of $\theta$:

$$[X^*_{\text{add}}, T_{\text{add}}] = \arg\max_{X^*_{\text{ca}}, T_{\text{ca}} \in \mathcal{D}_1, T_{\text{ca}} \in \mathcal{D}_2} - \sum_i \frac{\partial \mathcal{L}_{\text{ca}}(X^*_\text{ca}, T_{\text{ca}})}{\partial \log \xi_i},$$

where $\xi_i$ is the parameter of the of variational distribution controlling the variance of $\theta_i$. The function $\mathcal{L}_{\text{ca}}$ returns the lower bound in (2.14) after updating the variational distribution with candidate input points.

In the case of GPs with random features, the update of the variational distribution can be computed analytically (see, e.g., Section 2.3.3 of Bishop (2006) for a derivation), obtaining updated mean and covariance as

$$\mu_{\text{ca}} = \Sigma_{\text{ca}} \left( \Sigma_0^{-1} \mu_0 + \sigma_z^2 \Phi_{\eta, \text{ca}} z_{\text{ca}} \right),$$

$$\Sigma_{\text{ca}} = \sigma_z^2 \left( \Phi_{\eta, \text{ca}}^\top \Phi_{\eta, \text{ca}} + \sigma_z^2 \Sigma_0^{-1} \right)^{-1},$$

(5.2)
with $\Phi_{\eta,ca} \in \mathbb{R}^{N_{ca} \times N_{RF}}$ the row by row evaluation of $\phi_{\eta}(x,t)^\top$ on $X^*_{ca}$ and $T_{ca}$, $\mu_0$ and $\Sigma_0$ are the mean and the covariance of the variational distribution before selecting the candidate points, and $\sigma^2_{z} > 0$ corresponds to the noise term of the likelihood function. The criterion forecasts the effect of evaluating $\eta$ at $[X^*_{ca}, T_{ca}]$ using the predictive mean $z_{ca} = E(\eta(X^*_{ca}, T_{ca}) | U, \psi, \Omega)$. Therefore, the adaptive sampling chooses new input locations which maximally reduce the variance of the approximate posterior over $\theta$.

As an illustration of this approach, we perform adaptive sampling on the borehole function in (4.1), starting from $N_0 = 100$ points and adding batches of $N_{add} = 20$ until we reach $N = 400$. The initial design is obtained by a latin hypercube sampling with optimized maximin distance. The sampling criterion in (5.1) is optimized with gradient descent using automatic differentiation with respect to $X^*_{ca}$ and $T_{ca}$. We compare against latin hypercube sampling with optimization of the maximin distance, which is a classic space-filling design, for $N = 100, 200, 300$ and $400$. The experiment is reproduced five times with different initial and space filling designs, and we report calibration errors $||E_q(\theta) - \theta||$ with standard deviations in Figure 9. The designs produced by the adaptive method lead to consistently lower calibration error due to the optimized locations of the computer runs.

**Identifiability** The issue of identifiability that we discussed for the KOH model affects our formulation too. In particular, for the generalized formulation where the discrepancy is not modeled through an additive component but through a composition with an unknown function $\gamma(\eta(x,t), x)$, our approach yields a posterior over $\theta$ and $\gamma(\eta(x,t), x)$ which concentrates on the manifold

$$\mathcal{M} = \{(t, \gamma(\eta(x,t), x)) \mid \gamma(\eta(x,t), x) = f(x,t) \text{ with } x \in X \cup X^*\},$$

as the number of observations increases. Ignoring the discrepancy by removing the composition of the warping function with $\eta(x, \theta)$ would not constitute a problem from the implementation perspective, and our approach would result in a fast and flexible calibration model where $\theta$ is identifiable. Similarly to the KOH model, however, ignoring the discrepancy results in a model that could potentially yield a biased estimation of the parameters $\theta$, depending on how accurate $\eta(x, t)$ is in modeling the physical process.

The composition with a discrepancy function is therefore needed to avoid bias in the estimate of $\theta$. As a result, the importance of priors over $\theta$ and model discrepancy to mitigate the issue of lack of identifiability is as important as in the KOH model. This can dictate the choice of GP versus DGP priors. For instance, in the radionuclide diffusion test case (Section 4.3), a preview of the hectic observations hints in favor of a DGP model. On the contrary, in Section 4.4 knowing that the borehole function is smooth, the computer model and the discrepancy were given GP priors. Beyond these general considerations, there exist works allowing to impose constraints on DGPs priors, such as positivity, monotonicity, convexity, or being solutions to differential equations. In Lorenzi and Filippone (2018), this is done within the same framework proposed here involving random feature approximations and variational inference, so it would be rather straightforward to incorporate these in our model and inference.
Other ideas proposed in the literature on how to deal with the lack of identifiability could easily be adapted to our framework. For example, when multiple responses are available and they are mutually dependent on the same set of calibration parameters $\theta$, this improves identifiability (Arendt et al., 2012), and accommodating for multiple responses in our framework is straightforward. Furthermore, it would be possible to apply similar ideas to the ones proposed in Tuo and Wu (2016); Wong et al. (2017) where GP priors are replaced by DGP.

Another exciting line of research, which we could benefit from in order to investigate the identifiability properties of DGP-cal, follows recent results on so-called disentanglement (Locatello et al., 2019). This literature studies the properties of deep models in the context of density estimation, through the use of variational autoencoders (Kingma and Welling, 2014). Autoencoders are models composed of two elements: an encoder and a decoder. The encoder maps a set of observations into a low-dimensional latent representation, whereas the decoder maps the latent variables into the observations. In variational autoencoders, one is interested in obtaining a distribution over the latent variables by means of a variational formulation. Recently, a lot of interest has been devoted to disentangled representations, which are those where the latent variables are independently controlling different generative factors, and their number is sufficient to capture the diversity observed in the observations. A popular example is the one where observations are images of a ball of different colors and sizes, and with varying color of the background; the three generating factors are color and size of the ball, and color of the background, and one hopes to be able to infer this without any prior knowledge and simply by analyzing the given images. Locatello et al. (2019) made a theoretical breakthrough showing that this is impossible unless some extra information on these generating factors is available. Following this negative result, Khemakhem et al. (2020) provided theoretical guarantees on how to recover identifiability of these latent generating factors by means of a factorized prior distribution over the latent variables that is conditioned on some additional observed variables. While the setup is different to calibration, there are many similarities with our model in that they both use deep models and variational inference. Our calibration model can be seen as the decoder of such variational autoencoders, and $\theta$ in our model can be seen as the latent generative factors without encoder. We believe that this is an interesting direction to establish novel results and insights on the identifiability of the proposed calibration model.

References


DGPs for Calibration of Computer Models


Invited Discussion∗

Dave Higdon† and Stephen Walsh‡

Thanks to the authors for this enjoyable and interesting paper. From a narrow perspective, this paper adds some welcome advances to the field of Bayesian computer model calibration. From a broader perspective, this paper gives

• a compelling example of embedding deep Gaussian Processes (DGPs) within an encompassing hierarchical modeling framework, and

• an engineered inference scheme that thoughtfully combines (and trades off) statistical modeling, distributional approximation, computation, and the quantification of uncertainty.

Here we discuss these broad themes, as well as specifics regarding the computer model calibration problem.

This paper makes it clear that the idea of embedding DGPs within larger modeling frameworks is an important one and will likely be so for the foreseeable future. Embedding the model discrepancy within a DGP to account for the difference between computer model and system measurements is particularly innovative.

As algorithms and code for implementing DGP’s become more available, it will be more common to see DGPs as part of a larger modeling framework – as it is in this paper. For example, we see them in multifidelity models (Perdikaris et al., 2017; Cutajar et al., 2019); they also show up implicitly in non-stationary spatial models (Schmidt and O’Hagan, 2003) as well as in models for coupling computer models (Kyzyurova et al., 2018). We currently are working on hierarchical models where one of the layers of a DGP links related processes. As a community, I’m sure we’ll figure out the settings in which these more complex models can best aid in statistical inference.

This paper/framework is also an excellent example of thoughtfully combining probabilistic modeling, computation and approximation in the service of statistical inference. This work does a great job in leveraging variational approximations and random feature representations so that this framework can handle very high volumes of observation and simulation output.

We note that the statistician’s modeling and specification choices can influence even a seemingly fixed model formulation. For example, with the Kennedy and O’Hagan (2001) formulation one has the choice of how to standardize data, represent computer

∗This work was supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research and Office of High Energy Physics, Scientific Discovery through Advanced Computing (SciDAC) program under Award Number 0000231018.

†Department of Statistics, Virginia Tech, dhigdon@vt.edu

‡Department of Statistics, Virginia Tech, walsh124@vt.edu
model (and observation) output, priors that control the GP’s, priors for the model parameters, priors for the observation error, etc. This new DGP formulation puts even more choices and controls in the hands of the statistician. We think this is a good thing. But it puts more responsibility on the modeler and makes details of comparisons a bit tricky.

We suspect that a DGP emulator may be a good fit for the contaminant transport problem of Section 4.3. Some of the outputs are concentrations at different times. Here some model parameters control the speed of contaminant transport, moving the spike in concentration to later or earlier times (Figure 7). Such shifts over the support $x$ are not easily captured with standard GPs or additive tree models. However, a DGP that allows one of the latent layers to warp time as model parameters vary could be very well suited to the emulation task in this example.

With very large numbers of computer model runs, uncertainty in emulation often reduces to the point of being negligible in comparison to uncertainty in other features/parameters of the model. In such settings, a number of machine learning-based emulators can be quite advantageous – even when their uncertainty is only characterized approximately.

However, as mentioned in this paper and many others, uncertainty regarding model parameters and model discrepancy does not necessarily decrease with additional model runs and physical measurements. It has been our preference to embrace this joint uncertainty and incorporate it into the posterior inference. This indeterminacy is not a conceptual problem under the Bayesian paradigm (but some approximations may have difficulty accurately describing this joint parameter-discrepancy distribution). In new, more extrapolative settings this wider parameter uncertainty can play an important role in capturing prediction uncertainty. For example, some parameters (often governing material behavior) are not well constrained by initial experiments – especially while allowing for model discrepancy. But new experiments may explore more extreme temperature, pressure, or strain-rate regimes where such parameters are now influential. Admitting the full range of uncertainty allowed by previous experiments is crucial for producing useful prediction uncertainties. This is also crucial for planning future experimental campaigns. With all of this in mind, interpreting comparisons such as those in Fig 4 may not be straightforward. That said, we don’t have any better ideas at this stage.

 Appropriately capturing uncertainty also highlights the importance of specifying the prior discrepancy distribution. In the additive discrepancy setting, we have some experience and examples to lean on. Defining a prior discrepancy model that’s embedded within a DGP framework strikes us as something that’s a bit trickier. Do the authors have any thoughts or guidance in this regard?

Once again, our thanks to the authors for advancing the field: producing a self-contained package for handling computer model calibration based on a new DGP model formulation that’s flexible and capable of handling much larger datasets.


References


Invited Discussion

Anthony O’Hagan*

Reading this paper was truly delightful. Having played my part in starting this line of research by explicitly recognising the importance of model discrepancy in computer models, I am thrilled to see how it has moved on in the last twenty years. I will confess right away that I have not kept up with the technical and computational developments employed in this paper; I was aware of deep Gaussian processes and variational inference, but had no real appreciation of their mathematics and what they could do. It seems that they offer enormous advances in this field of computer model calibration.

Having said that, there remains the problem of identifiability. As the authors point out clearly in Section 5, we cannot hope to learn about meaningful, physical parameters without additional prior information. And while that may come in part from prior information about the parameters themselves, the enterprise has hardly achieved much if we only learn about them \textit{a posteriori} by having good knowledge of them \textit{a priori}. The crucial step is to incorporate prior knowledge of the model discrepancy, and it seems to me that while the developments in this paper resolve the principal computational challenges in the original Kennedy-O’Hagan formulation, they may do so at the expense of making it harder to see and construct the prior information. I strongly encourage the authors and other researchers in the field to demonstrate, in real applications, how genuine prior knowledge can be incorporated into a deep Gaussian process model for the discrepancy.

*University of Sheffield, a.ohagan@sheffield.ac.uk
Invited Discussion

Annie Sauer* and Robert B. Gramacy†

Let us begin by congratulating Marmin and Filippone (2022) for a timely and important paper. Their methodological advancements touch on several themes that are important to our own research – computer surrogate modeling at scale and at high fidelity, uncertainty quantification, and active learning for sequential design – and about which we shall offer some discussion below. Broadly, as advances in predictive modeling, primarily in machine learning (ML), work their way back to their statistical origins (e.g., deep Gaussian processes [DGPs] originated in the geo-spatial statistical literature (Sampson and Guttorp, 1992; Schmidt and O’Hagan, 2003)), one may naturally worry at what risk, or what has been lost? Marmin and Filippone do a fantastic job, in our opinion, of borrowing from ML for computer model calibration. In so doing, they demonstrate that it is possible to enjoy advances in modern analytics in the context of an endeavor demanding numerically cumbersome posterior inference and full uncertainty quantification (UQ): something dear to statisticians but often ignored in ML. Here we offer some thoughts on how (we hope) things can continue to move, productively, in that direction while at the same time shamelessly promoting some of our own work.

Diverging goals of calibration

In their seminal paper on Bayesian computer model calibration, Kennedy and O’Hagan (2001, KOH, hereafter) acknowledged that their framework would excel at synthesis (of computer model runs and field data), but struggle at identification of the calibration parameter (the inverse problem): \( \theta \). Without getting too technical, this is because the modeling apparatus – coupling two GPs, one for the computer model and one for the bias – together with \( \theta \) is “too flexible”. Since then, and especially recently, the literature has diverged. Some, beginning with Higdon et al. (2004) and Liu et al. (2009), and more recently with Brynjarsdóttir and O’Hagan (2014); Plumlee (2017); Gu and Wang (2018); Gu (2019); Tuo and Wu (2015, 2016); Wong et al. (2017); Plumlee (2019) focus on identification of \( \theta \) in the inversion problem, mostly by reigning in that flexibility. Others, like this discussion paper, but also including Konomi et al. (2017); Karagiannis et al. (2019); Cheng et al. (2021) and some of our own work (Gramacy et al., 2015; Huang et al., 2020; Huang and Gramacy, 2021) focus instead on modeling computer simulations at scale, enhancements in fidelity of the surrogate, and on synthesis of data sources (simulation and physical observation) for prediction and UQ. In other words, emphasizing more flexibility because an ordinary GP is insufficient when physical processes are nonstationary or exhibit discontinuity, for example.

This divide, between identification and modeling fidelity, reminds us of a similar tussle in econometrics and clinical trials: that between causal inference/treatment effects, and that of modeling for the purpose of reproducing behavior, and for accurately...

*Department of Statistics, Virginia Tech, anniees@vt.edu
†Department of Statistics, Virginia Tech
predicting the outcome (with UQ) for an experiment in novel circumstances. The two would seem to be fundamentally at odds with one another. Identification requires more things be “bolted down”, anathema to modern ML, whereas predictive accuracy and UQ require a delicate balance between fidelity and regularization – allowing the model to exhibit and learn complex dynamics when the data support it.

Our opinion is that in the context of computer surrogate modeling, you have to have the first (fidelity) before you have the second (identification). We think KOH would agree. If you can’t handle the scale of the simulation experiment (tens of thousands to millions of runs), and (in spite of such large data) you can’t capture the nuances of the (e.g., physical) dynamics in play, then it doesn’t matter if the theory says you can identify $\theta$ or not. Ideally you would have both, but at this time – and perhaps indefinitely if econometrics is any guide – that would seem to be illusive. We therefore congratulate Marmin and Filippone for making important advancements in modeling fidelity by deploying DGPs where ordinary (shallow) GPs are not enough. We wish they, and other authors before them, didn’t have to apologize for not accommodating identification (awkward paragraphs in Sections 2.1 and 5), and we hope that their paper will provide inspiration to continue with advances along modeling fidelity lines.

Chapter 8.1 of Surrogates (Gramacy, 2020) discusses a ball-drop experiment where the computer model incorporates acceleration due to gravity, but omits atmospheric resistance among other factors. By simplifying the KOH apparatus you might believe that you can identify $\theta$, the acceleration due to gravity. But in fact you can’t, not with any amount of data, not with any of the theoretically sound papers cited above. This is a common situation. You can only “invert $\theta$” modulo the fictional scenario that your computer model includes all dynamics exhibited in field. In practice your model will idealize some of those dynamics, possibly over-simplifying important ones (because all models are wrong). However, it is possible to accurately predict the ball drop time, accounting for both forces and anything else omitted from the computer model, if you use the full KOH apparatus because it does a great job of synthesizing information sources. The only limitation here is whether or not your class of models for the surrogate, and the discrepancy term, is rich enough to capture dynamics which the data (perhaps a large amount of it) reveal. This is where ML tools are most valuable, and where Marmin and Filippone make an important advance on the state-of-the-art.

**Surrogate modeling fidelity and UQ**

But at the same time, we wonder if the variational inference (VI) and random feature expansion (RFE) approach by Marmin and Filippone is really doing the best job at prediction and UQ. The combination of these boils down to two choices: an inferential scheme to handle the intractible DGP posterior (VI) and an approximation apparatus to circumvent the cubic costs of GP inference (RFE), which are exacerbated in a DGP setting. VI, although popular in the DGP literature, offers an approximate solution by settling for a “best guess” distribution from a known target family. By nature, this approximation oversimplifies the complexities of the DGP posterior. Bayesian Markov chain Monte Carlo (MCMC) sampling approaches, while computationally heavier, offer full posterior integration and thorough UQ. See Sauer et al. (2022b) for an ellipti-
cal slice sampling (ESS) approach and Havasi et al. (2018) for a Hamiltonian Monte Carlo (HMC) implementation. Until recently, the prevailing approximation apparatus for DGPs was via inducing points (Snelson and Ghahramani, 2006). Inducing points yield disappointingly blurry predictions; RFE is perhaps a promising alternative. We have found recent success with the Vecchia approximation (Vecchia, 1988) for fully Bayesian DGPs (Sauer et al., 2022a).

The pivotal question is, which of these offer the best DGP surrogate predictions/UQ at reasonable computational cost? The answer unfortunately is masked by coding considerations; existing codes combine one of each of the above tools (VI/MCMC and inducing points/RFE/Vecchia), along with varying treatment of kernel hyperparameters, etc., making it tricky to pull apart individual components. Nevertheless, a candid comparison is warranted. To this end, we offer a simulation study of current DGP surrogate implementations on two nonstationary test cases (from Surjanovic and Bingham, 2013). The DGP comparators we considered are: doubly stochastic VI + inducing points of Salimbeni and Deisenroth (2017) via gpflux (Dutordoir et al., 2021), HMC + inducing points of Havasi et al. (2018), ESS + Vecchia approximation of Sauer et al. (2022a) via deepgp (Sauer, 2022), VI + RFE of Cutajar et al. (2017), and VI + RFE of Marmin and Filippone (2022). Prediction error (root mean squared error [RMSE]) and UQ performance (continuous rank probability score [CRPS], lower is better) for each DGP surrogate are provided in Figure 1. Code to reproduce all results is available in our public repository.

Both VI + RFE implementations utilize the same approximation tools yet yield drastically different results, making it clear that coding and modeling choices can be as impactful as the choice of statistical tools. While it’s impossible to fully separate inferential schemes, approximation choices, and other coding considerations, several things are still clear. Sampling schemes (HMC + ESS) tend to outperform VI. In all but the smallest settings, inducing point and RFE variations appeared to saturate and were unable to improve learning from larger training data sizes. In contrast, the Vecchia approximation consistently improved as training data size increased. The combination of full posterior integration and high-fidelity Vecchia approximation (ESS + Vecchia) outperformed across the board.

DGP surrogates are growing in popularity and impact, but the existing public implementations are a bit of a Wild West. There is great potential for off-the-shelf DGP implementations that work well in robust settings. In surrogate settings where prediction accuracy and UQ are essential, full posterior integration is clearly superior to VI, albeit at the cost of some extra computation. VI may still have its place when data sizes are enormous (full posterior sampling for DGPs has only recently become accessible for data sizes in the hundred thousands, Sauer et al. (2022a)), but when data is abundant, the benefit of a DGP surrogate’s nonstationarity may be less impactful.

Active learning

As a final thought we wish to touch on Marmin and Filippone’s Section 5 discussion of potential for active learning (AL) in calibration contexts, i.e., of acquiring new com-
computer model runs or field data observations to improve the fit. UQ is essential to AL enterprises, and AL is essential when training data sizes are limited. This is especially the case for DGP surrogates, as remarked by Sauer et al. (2022b). If you have limited training data, then it is important to place runs – or learn where to place runs – so that you can learn how dynamics are changing in the input space. With ordinary GPs, AL isn’t much help because it invariably learns to space-fill the study region. But with DGPs the resulting designs can be highly non-uniform and, consequently, lead to superior predictive performance. We agree with Marmin and Filippone that it would be interesting to study this further in a calibration context.

References


Invited Discussion


