Uncertainty Propagation using Infinite Mixture of Gaussian Processes and Variational Bayesian Inference

Peng Chen\textsuperscript{a}, Nicholas Zabaras\textsuperscript{b,a}, Ilias Bilionis\textsuperscript{c}

\textsuperscript{a} Materials Process Design and Control Laboratory, Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 14853-3801, USA
\textsuperscript{b} Warwick Centre for Predictive Modelling, The University of Warwick, Coventry CV4 7AL, UK
\textsuperscript{c} School of Mechanical Engineering, Purdue University, 585 Purdue Mall, West Lafayette, IN 47906-2088, USA

Abstract

Uncertainty propagation in flow through porous media problems is a challenging problem. This is due to the high-dimensionality of the random property fields, e.g. permeability and porosity, as well as the computational complexity of the models that are involved. The usual approach is to construct a surrogate response surface and then use it instead of the expensive model to carry out the uncertainty propagation task. However, the construction of the surrogate surface is hampered by various aspects such as the limited number of model evaluations that one can afford, the curse of dimensionality, multi-variate responses with non-trivial correlations, potential localized features of the response and/or discontinuities. In this work, we extend upon the concept of the Multi-output Gaussian Process (MGP) to effectively deal with all of these difficulties simultaneously. This non-trivial extension involves an infinite mixture of MGP’s that is trained using variational Bayesian inference. Prior to observing any data, a Dirichlet process is used to generate the components of the MGP mixture. The Bayesian nature of the model allows for the quantification of the uncertainties due to the limited number of simulations, i.e., we can derive error bars for the statistics of interest. The automatic detection of the mixture components by the variational inference algorithm is able to capture discontinuities and localized features without adhering to ad hoc constructions. Finally, correlations between the components of multi-variate responses are captured.

\textsuperscript{*}Corresponding author: N. Zabaras, Email: nzabaras@gmail.com, URL: http://www.zabaras.com/
by the underlying MGP model in a natural way.

Key words: Variational Inference, Infinite Mixture of Gaussian Processes, Multi-output Gaussian Process, Nonparametric Variational Inference, Dirichlet Process, Reservoir Simulation, Uncertainty Quantification

1. Introduction

Reservoir simulation involving multiscale/multiphysics effects has been a central research topic during the past decades. Modern reservoir characterization and geostatistical modeling techniques aim at integrating information from different scales to build high resolution models with multi-million cells that describe the heterogeneous reservoir properties in great detail. Multi-phase flow through these highly detailed reservoirs is then studied using either Finite Element (FEM) or Finite Volume (FVM) methods. Typically, simulation times are prohibitively large for uncertainty propagation or inversion tasks. To decrease the computational cost, one resorts to multiscale methods [1, 2, 3]. These exploit the separation of scales and result in accelerated simulations.

Randomness in reservoir modeling is not intrinsic. That is, the permeability or porosity fields required for the construction of a reservoir are very specific physical quantities. However, we are unable to fully resolve these quantities experimentally. This is exactly why we are forced to treat them as uncertain. That is, the uncertainties involved represent a knowledge gap. The goal of uncertainty propagation is to quantify how this uncertainty propagates to the quantities of interest such as the pressure or the velocity fields.

In most studies of porous media flow, only the uncertainty of the permeability field is taken into account [4, 5]. The other important quantity, the porosity of the rock, is usually assumed to be constant. This is, partly, justified because of the uncertainty in porosity is at least one order of magnitude smaller than the uncertainty in permeability [6]. In addition, including both fields practically doubles the dimensionality of the uncertainty propagation problem. To the best of our knowledge, [7] is the first effort to simultaneously treat as random fields both permeability and porosity. In this work, we
do the same. However, we construct a realistic model that is based on data instead of a synthetic one.

There are two main difficulties associated with the representation of uncertainties in property fields for reservoir modeling. Firstly, getting subsurface experimental measurements is very expensive. Therefore, only a limited number of observations is available for this purpose. Secondly, both permeability and porosity fields are high-dimensional in nature. This mandates the use of a dimensionality reduction technique. Hence, the problem is to quantify the fields’ uncertainty based on a limited set of experimental observations with as few variables as possible. Towards this goal, we employ the well-known Karhunen-Loève expansion [8].

Given a characterization of the uncertainty in permeability and porosity, the next task is to propagate it through the model and characterize the uncertainty of the flow properties. We refer to this problem as the uncertainty propagation (UP) task. Monte Carlo (MC) is the classical approach to UP. It is extremely easy to implement and parallelize. However, convergence typically requires hundreds of thousands or even millions of samples. As a result, MC becomes inefficient for expensive models.

The Stochastic Finite Element Method (SFEM) [9] started a new wave of research in the field. In addition to the finite element basis, the response surface is also expanded in terms of orthogonal polynomials of the random variables (Polynomial Chaos (PC) or generalized PC (gPC) [10]). The coefficients of the PC expansion are found by solving an extended linear (or nonlinear) system derived from the partial differential equations of the model. The main drawback of the SFEM is that it scales badly with respect to the number the random variables.

Stochastic collocation methods extend upon the ideas of the SFEM, but do not require any intrusive changes to the simulator since they are approximating the PC coefficients by numerical integration. Stochastic collocation approaches based on PC, however, suffer from the curse of dimensionality, albeit they can deal with higher dimensions than SFEM. Another difficulty associated with any attempt to represent the response surface with PC, is the inability to capture sharp localized features and discontinuities. For low-dimensional settings, the Multi-Element gPC (ME-gPC) [11] has been successful in dealing with this issue. At higher dimensions (up to 70), Adaptive
Sparse Grid Collocation (ASGC) \cite{12} has been proven to be very effective. Gaussian processes \cite{13,14,15} are receiving significant attention in uncertainty quantification tasks even though they are limited by small datasets and covariance functions of moderate complexity. However, \cite{16} demonstrated that ASGC can be outperformed by a tree of Gaussian processes (tGP) when the number of simulations is low. Similar results were obtained in \cite{17} using a tree based on Relevance Vector Machines (RVM) with gPC basis functions. Despite their success, both \cite{16} and \cite{17} used ad hoc criteria for the adaptive construction of the trees. This aspect violates their Bayesian nature and necessarily leads to suboptimal results. From a completely different perspective, the authors of \cite{18,19} construct probabilistic graphical models representing the multiscale simulator. In this approach, the joint probability distribution of all coarse- and fine-scale variables is factorized locally in the graph following the structure defined by the underlying FEM mesh. This approach can deal with relatively high-dimensions while at the same time it can represent non-trivial correlations between the various variables. Stochastic coarse-graining arises naturally and a non-parametric implementation avoids the non-predictability issues typical of multiscale methods. In spite of its appeal, this approach requires complete data description of all random variables in the graph and remains computationally expensive and hard to implement. In addition, difficulties arise when applying it to surfaces with localized features or discontinuities.

The issue of limited data is best captured by the ideas outlined in \cite{5}. The authors use a fully Bayesian framework and are able to quantify the epistemic uncertainty induced by the limited number of simulations to the statistics of interest. The model they adopt for this purpose is the multi-output Gaussian process (MGP) of \cite{20}. The added benefit of using MGP is that it is able to capture the linear part of the correlations between distinct outputs (e.g., pressure and the velocity field) in a natural way. The use of a separable covariance function with a random, spatial, and time component makes possible to utilize linear algebra tricks that avoid the construction and inversion of large covariance matrices.

In order to be able to capture non-stationary effects such as localized features and discontinuities, we extend the MGP model in a non-trivial way. In particular, we consider an infinite mixture of MGP’s \cite{15}. Similar ideas have been used in various appli-
cation fields, e.g., Lázaro-Gredilla et al. [21] use a similar model to study a multi-target tracking problem, Ross and Dy [22] the lung disease sub-type identification problem, Yuan and Neubauer [23] robot kinematics, and Sun and Xu [24] a traffic flow problem. The first step in a mixture model, is to assign a label to each observation indicating to which one of the components of the mixture it belongs to. This label is treated as a latent random variable. Then, observations with the same label are grouped together in a single MGP model. To allow for an arbitrary number of mixture components, we make use of a Dirichlet process (DP) prior [25, 26, 27, 28, 29]. The posterior of all the parameters of the model is constructed by employing Variational Inference (VI) techniques [30, 24, 31]. VI techniques approximate the posterior by minimizing its Kullback-Leibler divergence (information loss) from a parametrized family of candidate distributions. In our case, we derive an approximation scheme that leads to a convergence rate that is orders of magnitude faster than traditional Markov Chain Monte Carlo (MCMC) sampling of the posterior. While there is no theoretical result that can access the performance and convergence of MCMC, the variational inference algorithm provides a clear way to assess its convergence by monitoring the values of the lower bound to the logarithm of the model evidence. This is one of the main reasons why we opted for the use of VI in this work. After approximating the posterior with VI, the UP problem can be solved by using the probabilistic surrogate. As in [5], the Bayesian nature of our model allows for the quantification of the uncertainty due to limited simulations.

The paper is organized as follows. In Section 2 we start by briefly presenting the core structure of our model and pay attention to each of the two main constituents: the MGP model (Section 2.1) and the DP prior (Section 2.2). In Section 2.4 we apply the VI methodology to our model and derive a fast approximation algorithm. Finally, in Section 2.5 we apply the constructed probabilistic surrogate surface to solve the UP problem. Numerical examples are presented in Section 3 demonstrating the high-accuracy and efficiency of the proposed framework. We conclude this work in Section 4.
2. Methodology

The response of a physical model can be represented as a multi-output nonlinear function, \( f : \mathcal{X} \rightarrow \mathbb{R}^q \), where \( \mathcal{X} \) denotes the input space and \( q \) is the number of output dimensions. The input space, \( \mathcal{X} \), usually contains three distinct components: one for stochastic inputs of the model, one for the spatial location, and one for time. The stochastic inputs is denoted by \( \xi \in \mathcal{X}_\xi \subset \mathbb{R}^{d_\xi} \), and are treated as uncertain. The spatial location and the time variables is denoted by \( s \in \mathcal{X}_s \subset \mathbb{R}^{d_s} \) with \( d_s = 1, 2 \) or \( 3 \) and \( t \in \mathcal{T} = [0, T] \) with \( T > 0 \), respectively. That is, the input space, \( \mathcal{X} \), is the Cartesian product:

\[
\mathcal{X} = \mathcal{X}_\xi \times \mathcal{X}_s \times \mathcal{T}.
\]  

(1)

For notational convenience, we refer to all the variables collectively by:

\[
x = (\xi, s, t).
\]  

(2)

The total dimensionality of the input space, \( \mathcal{X} \), is \( d = d_\xi + d_s + 1 \). The output, \( y \in \mathbb{R}^q \), of the response function, \( f(\cdot) \), at input \( x = (\xi, s, t) \),

\[
y = f(x) = f(\xi, s, t),
\]  

(3)

is thought as the value of the model at the spatial location \( s \) and time \( t \) when the stochastic inputs have the value \( \xi \). The number \( q \) of the distinct outputs depends on the model. For example, when modeling a dynamical system, \( q \) is equal to its dimension. For the 2D two-phase flow problem we consider, the distinct outputs are the pressure, the \( x \)-velocity, the \( y \)-velocity, and the saturation of the two phases, i.e., \( q = 4 \). This framework is applicable to most important problems.

In an UP problem, one assigns a probability density function (PDF) on the stochastic input, \( \xi \), and wishes to quantify its effect on the output, \( y \), at some specific spatial locations and time instants. We denote this PDF on \( \xi \) by \( p(\xi) \). This PDF usually represents our lack of knowledge about the true value of \( \xi \), rather than intrinsic randomness. Thus, the output of the model becomes a random field whose “randomness” corresponds to our uncertainty about the model’s predictions. The goal of UP is to
characterize the statistics of the output random field. Such statistics include the mean,

\[ m_f(s, t) = \int f(\xi, s, t) p(\xi) d\xi, \]  

(4)

the covariance,

\[ C_f(s, s', t, t') = \int \left( f(\xi, s, t) - m_f(s, t) \right) \left( f(\xi, s', t') - m_f(s', t') \right)^T p(\xi) d\xi, \]  

(5)

the PDF of the output at a particular spatial location \( s_0 \) and time instant \( t_0 \):

\[ p(y|s = s_0, t = t_0) = \int \delta(f(\xi, s_0, t_0) - y) p(\xi) d\xi, \]  

(6)

where \( \delta(\cdot) \) is Dirac’s delta function, and so on.

Because of the computational complexity associated with evaluating the model, \( f(\cdot) \), the statistics outlined above cannot be computed directly. To remedy this situation, we resort to the usual two-step procedure of most UP methodologies. First, we construct a surrogate of \( f(\cdot) \), and, subsequently, we use it for the calculation of the statistics. The surrogate construction is based on a set of observations \( D = (X, Y) \), where \( X = (x_1, \ldots, x_N)^T \in \mathbb{R}^{N \times d} \) and \( Y = (y_1, \ldots, y_N)^T \in \mathbb{R}^{N \times q} \) are the observed inputs and corresponding outputs, respectively. In particular, we consider a set of \( n_\xi \) observations of the stochastic inputs, \( \Xi = (\xi_1, \ldots, \xi_m) \), drawn from their PDF, \( p(\xi) \). For each observed \( \xi \), the model is evaluated on a preselected set of \( n_s \) spatial locations, \( S = (s_1, \ldots, s_N) \), at \( n_t \) time instants, \( t = (t_1, \ldots, t_n) \). That is, we have a total of \( N = n_\xi n_s n_t \) observations in \( D \) available for the surrogate construction.

Our surrogate is going to be a Gaussian process [14], albeit a non-trivial one. Therefore, it can be thought of as a Bayesian surrogate, i.e., a probability measure on the space of plausible models that is compatible with the data \( D \). As shown in [5], the Bayesian nature of the surrogate can be exploited to derive error bars for any computed statistic. These error bars correspond to the epistemic uncertainty induced by the limited amount of data in \( D \). Similar ideas, can be used to solve inverse problems with limited forward model evaluations [32]. This is one of the most important features of the Bayesian approach that makes it stand out from traditional UP methodologies. This point is further clarified in Section 2.5.

In an effort to capture non-stationary effects, such as localized features and discontinuities, we assume that the observed data are generated by an infinite number of
different latent MGP’s, \( \{ f^{(m)}(\cdot) \}_{m=1}^{\infty} \). Each latent MGP, \( f^{(m)}(\cdot) \), explains a subset of the data. We use a Dirichlet process (DP) prior to generate the components of the infinite MGP mixture prior to observing any data. In practice, we truncate this mixture to \( M \) components. Our objective has three parts: find the optimal number of latent functions needed to explain a given set of data, cluster each data point on one of these models, and train the \( m \)-th latent function, \( f^{(m)}(\cdot) \), using the data assigned to the \( m \)-th cluster.

The outline of the remaining of this Section is as follows. In Section 2.1 we introduce the MGP model used for each one of the latent functions, \( f^{(m)}(\cdot) \). The Dirichlet process is discussed in Section 2.2 and the variational inference algorithm for the model is presented in Section 2.4. Finally, Section 2.5 discusses the integration of the model with uncertainty quantification tasks.

### 2.1. Multi-output Gaussian process regression

Each of our latent functions, \( f^{(m)}(\cdot) \), is a MGP \([20, 5]\). In this section, we briefly outline the specifics of the MGP model. To keep the notation uncluttered, we are not using the label \( m \). However, every parameter in this section does implicitly depend on \( m \).

The prior measure on latent functions. Prior to seeing any data, we model the latent function, \( f(\cdot) \), as a \( q \)-dimensional Gaussian process:

\[
f(\cdot) | B, \Sigma, \theta \sim N_q( f(\cdot) | \mu(\cdot; B), c(\cdot, \cdot; \theta) \Sigma ),
\]

conditional on the hyper-parameters \( B, \Sigma, \) and \( \theta \). The symmetric, positive definite, correlation matrix, \( \Sigma \in \mathbb{R}^{q \times q} \), models the linear part of the correlations between the \( q \) distinct outputs. The mean function, \( \mu(\cdot; B) \), is given by:

\[
\mu(x; B) = B^T h(x),
\]

where \( h(\cdot) = (h_1(\cdot), \ldots, h_p(\cdot)) \) are regression functions common to all outputs and \( B \in \mathbb{R}^{p \times q} \). In the numerical examples we use linear regression functions:

\[
h(\cdot) = (1, x^T).
\]
The covariance function, \( c(\cdot, \cdot; \theta) \), is taken to be:
\[
c(x, x'; \theta) = \exp \left\{ -\frac{1}{2} \sum_{l=1}^{d} \left( \frac{(x_l - x'_l)^2}{(r_l)^2} \right) \right\} + \epsilon \delta(x - x'),
\]
where \( \theta = (r_1, \ldots, r_d, \epsilon) \in \mathbb{R}_{+}^{d+1} \). The parameter \( r_l \) can be interpreted as the length scale of the input dimension \( l, l = 1, \ldots, d \). The parameter \( \epsilon \) is known as the “nugget” and can be thought of as the variance of the noise. Equation (7) defines a probability measure on the space of surrogates that corresponds to our prior beliefs.

The prior of the parameters. For notational economy, we denote all the parameters by \( \phi \):
\[
\phi = (B, \Sigma, \theta).
\]
The parameters \( \phi \) take values in the space \( \Omega_\phi \):
\[
\Omega_\phi = \mathbb{R}^{p \times q} \times \mathbb{P}_q(\mathbb{R}) \times \mathbb{R}^{d+1},
\]
where \( \mathbb{P}_q(\mathbb{R}) \) is the space of \( q \)-dimensional positive-definite matrices with real coefficients. Following [20], we assign an uninformative prior on the pair \((B, \Sigma)\) of the form:
\[
p(B, \Sigma) \propto |\Sigma|^{-\frac{q+1}{2}},
\]
and, following [5], an exponential prior on each of the components of \( \theta \):
\[
p(\theta|\gamma) = \prod_{i=1}^{d} \mathcal{E}(r_i|\gamma_r) \mathcal{E}(\epsilon|\gamma_\epsilon),
\]
where \( \gamma = (\gamma_r, \gamma_\epsilon) \) are the hyper-parameters of the exponential distribution \( \mathcal{E}(\cdot|\gamma) \). The prior on \( \phi \) is given by the product rule assuming that \((B, \Sigma)\) and \( \theta \) are a priori independent:
\[
p(\phi|\gamma) = p(B, \Sigma)p(\theta|\gamma).
\]
The likelihood of the data. Now, assume that \( n \) observations have been classified to belong to the \( m \)-th cluster. Let \( X \) and \( Y \) be the corresponding inputs and outputs, respectively. We collectively denote them by \( D = (X, Y) \). The likelihood of \( Y \) under our model (Eq. (7)) is given by the matrix-normal [33]:
\[
Y|X, \phi \sim N_{n \times q}(Y|HB, A, \Sigma),
\]
where $H \in \mathbb{R}^{n \times p}$ is the design matrix,

$$H_{ij} = h_j(x_i), \quad (16)$$

and $A \in \mathbb{R}^{n \times n}$ is the covariance matrix,

$$A_{ij} = c(x_i, x_j; \theta). \quad (17)$$

The posterior of the parameters. The posterior of the parameters (conditioned on $\gamma$) is given by:

$$p(\phi|D, \gamma) = \frac{p(Y|X, \phi) p(\phi|\gamma)}{p(D|\gamma)}, \quad (18)$$

where $p(Y|X, \phi)$ is the likelihood given in Eq. (15) and $p(D|\gamma)$ is the evidence given by:

$$p(D|\gamma) = \int p(Y|X, \phi) p(\phi|\gamma) d\phi. \quad (19)$$

The posterior probability measure on latent functions. With the prior probability measure of Eq. (7) and conditioning on the observed data $D$, we derive the posterior probability measure on the space of surrogates:

$$f(\cdot)|\phi \sim N_q(f(\cdot)|\mu^*(\cdot; B), c^*(\cdot, \cdot; \theta) \Sigma), \quad (20)$$

where the posterior mean function is given by:

$$\mu^*(x; B) = B^T h(x) + (Y - HB)^T A^{-1} a(x), \quad (21)$$

and the posterior covariance function by:

$$c^*(x, x'; \theta) = c(x, x'; \theta) - a(x)^T A^{-1} a(x'), \quad (22)$$

with $a(\cdot) = (c(\cdot, x_1; \theta), \ldots, c(\cdot, x_n; \theta)) \in \mathbb{R}^n$. Eq. (20) can be used to draw samples of candidate surrogates that are compatible with the data $D$ as well as our prior beliefs if all the hyper-parameters are known. Alternatively, its mean function could be used as surrogate surface in the classical way.
Integrating out $B$ and $\Sigma$. If $n \geq p + q$ ($p$ is the number of regression functions and $q$ the number of output dimensions), then the choice of prior we made in Eq. (12) for the pair $(B, \Sigma)$ allows us to integrate it out from the posterior (see [20] for the details). In particular, multiplying Eq. (20) with Eq. (18) and integrating out $B$ and $\Sigma$, yields the posterior measure conditioned only on $\theta$ which is a $q$-dimensional student-$T$ process:

$$f(\cdot) \mid \theta, D \sim T_q(f(\cdot) \mid \mu^{**}(\cdot; \hat{B}), c^{**}(\cdot, \cdot; \hat{\Sigma}, n - p)).$$

(23)

with the mean and covariance functions:

$$\mu^{**}(x) = \mu^*(x; \hat{B}),$$

(24)

$$c^{**}(x, x'; \theta) = c^*(x, x'; \theta) + \left( h(x) - H^T A^{-1} a(x) \right)^T \left( H^T A^{-1} H \right)^{-1} \left( h(x') - H^T A^{-1} a(x') \right),$$

(25)

where

$$\hat{B} = \left( H^T A^{-1} H \right)^{-1} H^T A^{-1} Y,$$

(26)

$$\hat{\Sigma} = \frac{1}{n - p} \left( Y - H \hat{B} \right)^T A^{-1} \left( Y - H \hat{B} \right).$$

(27)

This approach is used for making predictions with each one of our latent models.

The posterior of $(B, \Sigma)$ conditioned on $\theta$. The posterior of $(B, \Sigma)$ has an analytic form:

$$p(B, \Sigma \mid D, \theta) = N_{p \times q} \left( B \mid \hat{B}, (H^T A^{-1} H)^{-1}, \Sigma \right) W_q^{-1}(\Sigma(n - p) \hat{\Sigma}, n - p),$$

(28)

where $W_q^{-1}(\cdot \mid W, \nu)$ is the Inverse-Wishart distribution [34] with scale matrix $W \in P_q(\mathbb{R})$ and $\nu$ degrees of freedom. This result is useful in the variational formulation because it specifies the form that a candidate posterior for the pair $(B, \Sigma)$ should take, as discussed in Section 2.4.5. The derivation of Eq. (28) is given in Appendix A.

2.2. The Dirichlet process

A DP [35] defines a probability measure on a space of probability measures. In particular, let $\Omega$ be a set and $\mathcal{F}$ be a $\sigma$-algebra on $\Omega$. The space of probability measures on $(\Omega, \mathcal{F})$ is denoted by:

$$\mathcal{P}(\Omega, \mathcal{F}) = \{ G : \mathcal{F} \to \mathbb{R}_+ \text{ is a probability measure} \}. $$

(29)
A DP defines a probability measure on $P(\Omega, F)$. That is, a sample $G(\cdot)$ from a DP is in $P(\Omega, F)$.

Let $G_0(\cdot) \in P(\Omega, F)$ and $\alpha_0 > 0$ some constant. Then $G(\cdot)$ is a sample from the DP induced by $G_0(\cdot)$ and $\alpha_0$, i.e.,

$$G(\cdot) \sim DP (G_0(\cdot), \alpha_0),$$

if and only if for any partition $\{A_1, \ldots, A_k\} \subset F$ of $\Omega$ we have:

$$(G(A_1), \ldots, G(A_k)) \sim \text{Dir}_k (G(A_1), \ldots, G(A_k)|\alpha_0 G_0(A_1), \ldots, \alpha_0 G_0(A_k)),$$

where $\text{Dir}_k(\cdot, \ldots, \cdot|\alpha_1, \ldots, \alpha_k)$ denotes the Dirichlet distribution with parameters $\alpha_1, \ldots, \alpha_k$.

The “stick-breaking” construction. The “stick-breaking” construction of [36] allows us to define the DP in a generative way using some intermediate random variables. In particular, let $G_0(\cdot)$ and $\alpha_0$ be as before and define the random variables $\nu = (\nu_1, \nu_2, \ldots) \in [0, 1]^\infty$ by:

$$\nu|\alpha_0 \sim \prod_{m=1}^{\infty} \text{Beta} (\nu_m|1, \alpha_0),$$

the sequence of numbers:

$$\pi_m(\nu) = \nu_m \prod_{i=1}^{m-1} (1 - \nu_i),$$

and the random variables $\omega = (\omega_1, \omega_2, \ldots)$ by:

$$\omega_m|G_0(\cdot) \sim G_0(\omega_m).$$

Then, the random probability measure on $(\Omega, F)$ defined by:

$$G(\cdot; \nu, \omega) = \sum_{m=1}^{\infty} \pi_m(\nu) \delta_{\omega_m}(\cdot),$$

where $\delta_{\omega_m}(\cdot)$ is Dirac’s delta function centered at $\omega_m$, is a sample from the DP defined in Eq. (30).

2.3. An infinite mixture of MGP’s using the Dirichlet Process

According to the discussion of Sec. 2.1, each latent MGP model is characterized uniquely by choosing its parameters $\phi \in \Omega_\phi$. The DP concept is used to define a prior
probability measure on the space of probability measures of the model space $\Omega_\phi$. The role of this DP is to generate the components of the MGP mixture prior to observing any data.

Let $\alpha_0$ be as before, $\mathcal{T}_\phi$ be a $\sigma$-algebra on $\Omega_\phi$, and $P_{\phi,0}(\cdot|\gamma)$ be the probability measure on $(\Omega_\phi, \mathcal{T}_\phi)$ induced by the prior $p(\phi|\gamma)$ of Eq. (14), i.e. for each $A \in \mathcal{T}_\phi$ we have:

$$P_{\phi,0}(A|\gamma) = \int_A p(\phi|\gamma) d\phi.$$  \hfill (36)

The Dirichlet process $DP(\cdot|P_{\phi,0}(\cdot;\gamma), \alpha_0)$ is used to define a prior probability measure on $P(\Omega_\phi, \mathcal{T}_\phi)$. A sample $P_{\phi}(\cdot)$ from $DP(\cdot|P_{\phi,0}(\cdot;\gamma), \alpha_0)$ is then used to generate the mixture components. Let $v$ and $\pi_m(v), m = 1, 2, \ldots$ be as in Eq. (32) and Eq. (33), respectively, and, as in Eq. (34), let

$$\Phi = (\phi_1, \phi_2, \ldots)$$ \hfill (37)

be generated by:

$$\phi_m|\gamma \equiv \phi_m|P_{\phi,0}(\cdot;\gamma) \sim P_{\phi,0}(\phi_m|\gamma) \equiv p(\phi_m|\gamma).$$ \hfill (38)

Notice that, because the measure $P_{\phi,0}$ is absolutely continuous (see Eq. (36)), it is guaranteed that all the $\phi_m$'s generated in Eq. (38) are distinct. Making use of the “breaking-stick” construction, we see that a sample from $DP(\cdot|P_{\phi,0}(\cdot;\gamma), \alpha_0)$ can now be represented (see Eq. (35)) as:

$$P_{\phi}(\cdot; v, \Phi) = \sum_{m=1}^{\infty} \pi_m(v) \delta_{\phi_m}(\cdot).$$ \hfill (39)

Each observation $(x, y)$ is generated by sampling one of the models $m = 1, \ldots, \infty$. Each model $m$ is uniquely characterized by the parameters $\phi_m$. These parameters need to be sampled from Eq. (39) (the sample from the GP). Because Eq. (39) is degenerate, we necessarily obtain one of the $\phi_m$'s we draw from the prior in Eq. (38). Thus to assign a model to $(x, y)$, one needs to simply pick a number $m = 1, \ldots, \infty$ by sampling the multinomial distribution with probabilities $\pi_m(v)$. Rather than doing this directly, is more convenient to introduce a latent (indicator) variable $z$ whose sole role is to pick a number from $m = 1, \ldots, \infty$ with the right probabilities (Eq. (39)). Then, if we draw a
z, we can say that \( \phi_z \) (the \( \phi_m \)'s are coming from Eq. (38)) is the model associated with \((x, y)\).

For \(N\) observations, we thus introduce indicator variables \(z_i \in \mathbb{N}\), for each \(i = 1, \ldots, N\). Let \(z \in \mathbb{N}^N\) be the vector of indicator variables, called the indicator vector:

\[
z = (z_1, \ldots, z_N).
\]

The prior assigned to \(z\) (following Eq. (39)) is taken as:

\[
p(z|\nu) = \prod_{n=1}^{N} \sum_{m=1}^{\infty} \pi_m(\nu) \mathbb{1}_{[z_n=m]} = \prod_{n=1}^{N} \text{Multi}(z_n|\nu_1, \nu_2, \ldots) = \prod_{n=1}^{N} \pi_{z_n}(\nu),
\]

where \(\text{Multi}(\cdot|\nu_1, \nu_2, \ldots)\) stands for the multinomial probability distribution with probabilities \(\pi_1, \pi_2, \ldots\). Observe that there are not necessarily \(N\) distinct latent models as \(\phi^*_n = \phi_{z_n}\) are distributed according to \(P_{\phi(\cdot; \nu, \Phi)}(\cdot)\) of Eq. (39).

Following [15], we consider the indicator vector to be dependent on the input:

\[
p(z|x, m, R, \nu) = \prod_{n=1}^{N} \prod_{m=1}^{\infty} \left( \frac{p(x_n|m, R_m)\pi_m(\nu)}{\sum_{j=1}^{\infty} p(x_n|m_j, R_j)\pi_j(\nu)} \right)^{\mathbb{1}_{[z_n=m]}},
\]

where \(\pi_m(\nu)\) is the prior on \(z\) defined above in Eq. (33) and \(p(x_n|m, R)\) is defined by a Gaussian as [24]:

\[
p(x_n|z_n = m, m, R_m) = \mathcal{N}(x_n|m, R_m^{-1}),
\]

with \(m, R_m\) its mean and precision matrix, respectively. We denote \(m = \{m_1, m_2, \ldots\}\) and \(R = \{R_1, R_2, \ldots\}\) the mean and precision for all mixture components. Conjugate priors to \(m\) and \(R\) are assigned as

\[
m = \mathcal{N}(\mu_0, R_0^{-1}),
\]

\[
R = \mathcal{W}(W_0, \nu_0).
\]

**Remark 1.** In addressing the input clustering problem, we consider a full precision matrix \(R_m\) that governs the size and shape of each local model \(m\). With such a choice, both the inner- and inter-correlations between the different input dimensions \((\xi, s, t)\)
are considered. A simplification of the clustering model above can be introduced by considering a diagonal precision matrix $R_m$ with each diagonal element related to the correlation length parameters of the MGP model (Eq. (9)). Alternative approaches for introducing dependence of the mixture weights on the covariates include the order-based dependent Dirichlet processes [37] and the local Dirichlet process in [38]. However, these approaches do not appear promising for the high-dimensionality problems of interest.

To finalize the description of the model, we define an operator that looks at $z$ and clusters the data $D = (X, Y)$ according to the latent function that explains them. Towards this goal, define the selection operator for any matrix $W \in \mathbb{R}^{n \times \ell}$ by:

$$S_m(W; z) := \{w_n : \text{if } z_n = m\},$$

where $w_n$ is the $n$-th row of $W$. $S_m(W|z)$ is to be thought of as a matrix with $|z_n = m|$ rows and $\ell$ columns ($|B|$ is the number of elements of the set $B$). Using this definition, the inputs and outputs associated with the latent function $m$ are $S_m(X; z)$ and $S_m(Y; z)$, respectively.

The likelihood of the full model can be described as:

$$p(Y, z|X, \Phi) = p(Y|X, z, \Phi) p(z|X, m, R, \nu),$$

where $p(Y|X, z, \Phi)$ is given as:

$$p(Y|X, z, \Phi) = \prod_{m: |\{z_n = m\}| > 0} p(S_m(Y; z)|S_m(X; z), \phi_m),$$

where the product is over latent functions that are associated with at least one observation. The likelihood term pertaining to the $m$-th latent function, $p(S_m(Y; z)|S_m(X; z), \phi_m)$, is given by Eq. (15) with $S_m(X; z), S_m(Y; z)$, and $\phi_m$ instead of $X, Y,$ and $\phi$, respectively.

The prior of the full model is:

$$p(\Phi, \nu, m, R|I) = p(\Phi|\gamma) p(\nu|\alpha_0) p(m|u_0, R_0) p(R|W_0, \nu_0),$$

where we define the hyper-parameters as $I = (\gamma, \alpha_0, u_0, R_0, W_0, \nu_0)$. $p(\Phi|\gamma)$, and $p(\nu|\alpha_0)$ are given by Eqs. (38) and (32), respectively, and $p(m|u_0, R_0)$ and $p(R|W_0, \nu_0)$ are given by Eqs. (44) and (45), respectively.
The posterior of the full model is:

\[
p(\Phi, z, \nu, m, R|\mathcal{D}, \mathcal{I}) = \frac{p(Y|X, \Phi) p(z|X, m, R, \nu) p(\Phi, \nu, m, R|\mathcal{I})}{p(\mathcal{D}|\mathcal{I})},
\]

where \( p(Y|X, \Phi) \), \( p(z|X, m, R, \nu) \) and \( p(\Phi, \nu, m, R|\mathcal{I}) \) are given in Eqs. (48), (42), and (49), respectively, and the evidence is:

\[
p(\mathcal{D}|\mathcal{I}) = \sum_z \int p(Y|X, \Phi) p(z|X, m, R, \nu) p(\Phi, \nu, m, R|\mathcal{I}) d\Phi d\nu dm dR.
\]

A probabilistic graphical model representation of this model is given in Fig. 1.

Figure 1: Graphical model representation of the introduced framework. The input variable \( x_n \) is determined by \( \xi, S \) and \( t, x_n \) and \( y_n, n = 1, \ldots, N \) are the observations. \( f^{(m)} \) denotes the \( m \)-th MGP model, \( \nu_m, \beta^{(m)}, B^{(m)} \), and \( \Sigma^{(m)} \) are the affiliated parameters to \( f^{(m)} \), and \( \alpha, \gamma_r, \gamma_c \) are the hyperparameters. \( m_n \) and \( R_m \) are parameters used to determine the clustering of observations, and \( u_0, R_0, W_0, \) and \( \nu_0 \) are the corresponding hyperparameters. \( z_n \) is the hidden variable that classifies each observation. Based on the classification, \( f^{(m)} \) is constructed using only the \( m \)-th data subset.

### 2.4. Variational inference

VI aims at obtaining an approximation of the posterior (see Eq. (50)) by transforming the inference problem to an optimization problem. In particular, it involves mini-
mizing the “distance” between the true posterior of Eq. (50) and a family of candidate probability distributions.

To simplify the notation, we denote both the latent variables $z$ and $\nu$, the input parameters $m$, $R$, and the parameters of the model $\Phi$ as follows:

$$\Psi = \{z, \nu, m, R, \Phi\}. \quad (52)$$

We seek an approximation $q(\Psi)$ to the true posterior of the model $p(\Psi|D, I)$ (see Eq. (50)) by minimizing the Kullback–Leibler (KL) divergence:

$$\text{KL}[q \parallel p] = \int q(\Psi) \ln \frac{q(\Psi)}{p(\Psi|D, I)} d\Psi. \quad (53)$$

The KL divergence can be thought of as a “distance” between two probability distributions. It is always greater than or equal to zero and is exactly zero when the two distributions coincide. Dealing with the posterior directly is impossible because the evidence $p(D|I)$ of Eq. (51) is not known. However, notice that (see Appendix B):

$$\ln p(D|I) = \text{KL}[q \parallel p] + L[q], \quad (54)$$

where $L[q]$ is given by:

$$L[q] = \int q(\Psi) \ln \frac{p(D, \Psi|I)}{q(\Psi)} d\Psi. \quad (55)$$

$L[q]$ is a lower bound to the logarithm of the evidence and depends only on the joint distribution of $\Psi$ and $D$, $p(D, \Psi|I)$. Therefore, minimizing Eq. (53) is equivalent to maximizing $L[q]$. Solving this maximization problem is the goal of the remaining of this section.

The joint distribution $p(D, \Psi|I)$ is represented as:

$$p(D, \Psi|I) = p(Y|X, z, \Phi) p(z|X, m, R, \nu) p(\Phi, \nu, m, R|I)$$

$$= \prod_{m=1}^{\infty} \left\{ p(S_m (Y; z) | S_m (X; z), \phi_m) \prod_{n=1}^{N} \left( \frac{p(x_n|m_n, R_n) \pi_m (y)}{\sum_{j=1}^{\infty} p(x_n|m_j, R_j) \pi_j (y)} \right) \right\} p(\Phi, \nu, m, R|I)$$

$$= \prod_{m=1}^{\infty} \left\{ p(S_m (Y; z) | S_m (X; z), \phi_m) \prod_{n=1}^{N} \left( \frac{p(x_n|m_n, R_n) \pi_m (y)}{C_n(m, R, \nu)} \right) \right\} p(\Phi, \nu, m, R|I). \quad (56)$$

17
where we define the normalization constant $C_n(m, R, \nu) = \sum_{j=1}^{\infty} p(x_n|m_j, R_j)\pi_j(\nu)$. Notice that the normalization constant depends on the parameters $m, R,$ and $\nu$. It contributes to the variational updates of the posterior distributions. Inspired by [31], we use $\hat{C}_n(\hat{m}, \hat{R}, \hat{\nu})$ to approximate $C_n(m, R, \nu)$, where $\hat{m}, \hat{R},$ and $\hat{\nu}$ are the expected values of $m, R,$ and $\nu$, respectively, over the corresponding posterior distributions. In practice, these values are obtained from the approximated posterior distributions of the previous iteration step of the variational inference algorithm.

Remark 2. Another possible way of approximating this normalization term is by finding an upper bound of it [39]. In the context of mixture of expert models, these bounds have been shown to allow straightforward implementation of variational inference for computing approximations to the posterior of the model parameters. However, application of such bounds to our infinite mixture of MGP models using Eq. (42) (note the presence of the $\pi_m(\nu)$ term inside the summation in the normalization factor) leads to non-standard distributions that can only be approximated by sampling or non-parametric (Gaussian mixture) forms. In addition to the unknown parameters $m, R,$ these bounds depend in a complicated manner on $\nu$. Thus, they do not lead to closed form approximations or a computationally efficient implementation.

Following [24, 27], we are looking for solutions that factorize as follows:

$$q(\Psi) = N \prod_{n=1}^{N} q(z_n) \prod_{m=1}^{M} q(\nu_m) \prod_{m=1}^{M} q(m_m) \prod_{m=1}^{M} q(R_m) \prod_{m=1}^{M} q(B_m, \Sigma_m) q(\theta_m).$$

(57)

Here, $M$, is the maximum number of models that we would like to be able to represent. This introduces a truncation level in the stick-breaking representation of the underlying DP. It must be noted at this point, that $M$ does not alter the infinite mixture prior. It is just part of the variational approximation of the true posterior and required to make the underlying calculations feasible.

Now, we proceed by iteratively maximizing Eq. (55) with respect to each one of the factors of Eq. (57) until a self-consistent solution is found. In particular, for any $\omega \in F_\varphi = \{ z_1, \ldots , z_n , \nu_1, \ldots , \nu_M , m_1, \ldots , m_M, R_1, \ldots , R_M, (B_1, \Sigma_1), \ldots, (B_M, \Sigma_M), \theta_1, \ldots, \theta_M \}$, the update equation for $q(\omega)$ is given as:

$$\ln q(\omega) = \mathbb{E}_{F_\varphi, I} [\ln p(\Psi, D|I)] + \text{const},$$

(58)

18
Thus, we can write:
\[ q_{m+1} = q_m + \frac{1 - \nu_m}{\nu_m} (x - \hat{m}_m) \]

where the “const” denotes the normalization factor to the corresponding distribution, and the expectation is with respect to \( q(\Psi) \) over all variables in \( F \) except \( \omega \). For a proof of Eq. (58), see Appendix C. All of these updates have an analytical form except the ones involving \( \theta_m, m = 1, \ldots, M \). In what follows, we derive all the updates. In addition, we show how to perform the \( \theta_m \) updates in an approximate, non-parametric manner.

2.4.1. Update of \( \nu_m \)

To compute the update of \( \nu_m \), we start with Eq. (58), use Eq. (56) and keep only the terms that depend on \( \nu_m \):
\[
\ln q(\nu_m) = \ln p(\nu_m|\alpha_0) + \sum_{n=1}^{N} \mathbb{E}_{F_n \setminus \nu_m}[\ln p(z_n|X, m, R, \nu)] + \text{const},
\]
where using Eqs. (33) and (41)
\[
\mathbb{E}_{F_n \setminus \nu_m}[\ln p(z_n|X, m, R, \nu)] = \mathbb{E}_{F_n \setminus \nu_m}\left[ \ln \left( \frac{p(X|n, m, R)}{C_n(m, R, \nu)} \right) \right]
\]
\[
= \mathbb{E}_{F_n \setminus \nu_m}\left[ q(z_n = m) \ln \nu_m + \sum_{i=m+1}^{M} q(z_n = i) \ln (1 - \nu_m) \right] + \text{const}
\]
\[
= \mathbb{E}_{C_n}[q(z_n = m)] \ln \nu_m + \mathbb{E}_{C_n}[q(z_n > m)] \ln (1 - \nu_m) + \text{const}.
\]

Note here \( \mathbb{E}_{C_n}(m, R, \nu) \) is calculated using the mean of the approximated posterior distribution from the previous iteration step, i.e. it is taken as constant that does not contribute to the update of \( q(\nu_m) \).

Finally using Eq. (32):
\[
\ln p(\nu_m|\alpha_0) = (\alpha_0 - 1) \ln (1 - \nu_m).
\]

Thus, we can write:
\[
\ln q(\nu_m) = (\alpha_0 - 1) \ln (1 - \nu_m) + \sum_{n=1}^{N} [\mathbb{E}_{C_n}[q(z_n = m)] \ln \nu_m + \mathbb{E}_{C_n}[q(z_n > m)] \ln (1 - \nu_m)] + \text{const}
\]
\[
= \left[ \sum_{n=1}^{N} \mathbb{E}_{C_n}[q(z_n = m)] \right] \ln \nu_m + \left[ \alpha_0 - 1 + \sum_{n=1}^{N} \mathbb{E}_{C_n}[q(z_n > m)] \right] \ln (1 - \nu_m) + \text{const}.
\]

From the above equation, we conclude that \( q(\nu_m) \) follows a Beta distribution:
\[
\nu_m \sim \text{Beta} \left( 1 + \sum_{n=1}^{N} \mathbb{E}_{C_n}[q(z_n = m)], \alpha_0 + \sum_{n=1}^{N} \mathbb{E}_{C_n}[q(z_n > m)] \right) \quad (60)
\]
2.4.2. Update of \( q(m) \)

To update of \( q(m) \), again, we use Eq. (58) and Eq. (56) and keep only the terms that depend on \( m \), so we have:

\[
\ln q(m) = \ln p(m|u_0, R_0) + \sum_{n=1}^{N} E_{F_n|m_n} [\ln p(z_n|X, m, R, \nu)] + \text{const}.
\]  
(61)

The first prior term gives:

\[
\ln p(m|u_0, R_0) = -\frac{1}{2} (m - u_0)^T R_0 (m - u_0) + \text{const}.
\]  
(62)

The second term can be calculated as:

\[
E_{F_n|m_n} [\ln p(z_n|X, m, R, \nu)] = E_{F_n|m_n} \left[ \ln \left( \prod_{i=1}^{M} \left( \frac{p(x_n|m_i, R_i)\nu_i(\nu)}{C_{\nu_i}(m_i, R_i, \nu)} \right)^{1[z_n=i]} \right) \right]
\]

\[
= E_{F_n|m_n} \left[ \ln \left( \prod_{i=1}^{M} p(x_n|m_i, R_i)^{1[z_n=i]} \right) \right] + \text{const}
\]

\[
= \sum_{j=1}^{M} [\mathbb{E}_{z_n}[q(z_n = i)] E_{F_n|m_n} [\ln p(x_n|m_i, R_i)]
\]

\[
= \mathbb{E}_{z_n}[q(z_n = m)] E_{F_n|m_n} [\ln p(x_n|m_m, R_m)]
\]

\[
= \mathbb{E}_{z_n}[q(z_n = m)] \left( \frac{1}{2} (x_n - m_m) \right)^T E_{R_m}[R_m] (x_n - m_m) + \text{const}.
\]

Substituting in Eq. (61), we obtain:

\[
\ln q(m) = \frac{1}{2} (m - u_0)^T R_0 (m - u_0)
\]

\[
- \frac{1}{2} \sum_{n=1}^{N} [\mathbb{E}_{z_n}[q(z_n = m)] (x_n - m_m)^T E_{R_m}[R_m] (x_n - m_m)] + \text{const}
\]

\[
= \frac{1}{2} \left( m_m^T R_m m_m - 2 m_m^T R_m u_0 + \sum_{n=1}^{N} \mathbb{E}_{z_n}[q(z_n = m)] m_m^T E_{R_m}[R_m] m_m \right)
\]

\[
- 2 \sum_{n=1}^{N} \mathbb{E}_{z_n}[q(z_n = m)] m_m^T E_{R_m}[R_m] x_n.
\]

For notational convenience, let us denote \( R_{m1} = \sum_{n=1}^{N} \mathbb{E}_{z_n}[q(z_n = m)] E_{R_m}[R_m] \) and \( R_{m2} = \sum_{n=1}^{N} \mathbb{E}_{z_n}[q(z_n = m)] E_{R_m}[R_m] x_n \). It can easily be shown now that \( q(m) \) follows a Gaussian distribution:

\[
m_m \sim \mathcal{N}_d \left( u_m, (R_0 + R_{m1})^{-1} \right),
\]  
(63)

where \( u_m = (R_0 + R_{m1})^{-1} (R_0 u_0 + R_{m2}) \).
2.4.3. Update of q(R_m)

To update of q(R_m), we similarly keep only terms that are dependent on R_m, thus obtaining:

\[
\ln q(R_m) = \ln p(R_m|W_0, v_0) + \sum_{n=1}^{N} \mathbb{E}_{\psi_n|R_m} [\ln p(z_n|X, m, R, \nu)] + \text{const}. \tag{64}
\]

The first prior term gives:

\[
\ln p(R_m|W_0, v_0) = \frac{v_0 - d - 1}{2} \ln |R_m| - \frac{1}{2} \text{tr} \left( R_m W_0^{-1} \right) + \text{const}. \tag{65}
\]

The expectation term can be calculated as:

\[
\begin{align*}
\mathbb{E}_{\psi_n|R_m} [\ln p(z_n|X, m, R, \nu)] &= \mathbb{E}_{\psi_n|R_m} \left[ \ln \left( \prod_{i=1}^{M} \left( \frac{p(x_n|m_i, R_i)p_i(\nu)}{C_n(m, R, \nu)} \right)^{1[z_n=i]} \right) \right] \\
&= \mathbb{E}_{\psi_n|R_m} \left[ \ln \left( \prod_{i=1}^{M} p(x_n|m_i, R_i)^{1[z_n=i]} \right) \right] + \text{const} \\
&= \mathbb{E}_{\psi_n} [q(z_n = m)] \mathbb{E}_{m_n} [\ln p(x_n|m_n, R_m)] + \text{const} \\
&= \frac{1}{2} \mathbb{E}_{\psi_n} [q(z_n = m)] \left( \ln |R_m| - \mathbb{E}_{m_n} (x_n - m_n)^T R_m (x_n - m_n) \right) + \text{const}.
\end{align*}
\]

Since \((x_n - m_n)^T R_m (x_n - m_n)\) is a scalar, we can write:

\[
\text{tr} \left( (x_n - m_n)^T R_m (x_n - m_n) \right) = \text{tr} \left( R_m (x_n - m_n)(x_n - m_n)^T \right), \tag{66}
\]

and thus simplify as:

\[
\begin{align*}
\mathbb{E}_{\psi_n|R_m} [\ln p(x_n|z_n, m_n, R_n)] &= \mathbb{E}_{\psi_n} [q(z_n = m)] \left( \ln |R_m| - \text{tr} \left( R_m \mathbb{E}_{m_n} (x_n - m_n)(x_n - m_n)^T \right) \right) + \text{const}.
\end{align*}
\]

Finally, we can write:

\[
\begin{align*}
\ln q(R_m) &= \sum_{n=1}^{N} \mathbb{E}_{\psi_n} [q(z_n = m)] + v_0 - d - 1 \ln |R_m| \\
&\quad - \frac{1}{2} \text{tr} \left( R_m \left( W_0^{-1} + \sum_{n=1}^{N} \mathbb{E}_{\psi_n} [q(z_n = m)] \mathbb{E}_{m_n} (x_n - m_n)(x_n - m_n)^T \right) \right) + \text{const}.
\end{align*}
\]

Thus R_m follows a Wishart distribution:

\[
R_m \sim \mathcal{W}_d(W_m, \nu_m), \tag{67}
\]
where

\[
\nu_m = \sum_{n=1}^{N} \mathbb{E}_{z_n}[q(z_n = m)] + \nu_0, \tag{68}
\]

\[
W_m^{-1} = W_0^{-1} + \sum_{n=1}^{N} \mathbb{E}_{z_n}[q(z_n = m)]\mathbb{E}_{m_n}[\{(x_n - m_m)(x_n - m_m)^T\}] \tag{69}
\]

2.4.4. Update of \( q(z_n) \)

Similarly to find the update of \( q(z_n) \), we start with Eq. (58), use Eq. (56) and keep only the terms that depend on \( z_n \):

\[
\ln q(z_n) = \mathbb{E}_{x_n|z_n} \left[ \ln p(z_n|X, \mu, \Sigma, \nu) + \ln p(y_n|x_n, z_n, \Phi, \mathcal{D}) \right] + \text{const}
\]

\[
= \mathbb{E}_{x_n|z_n} \left[ \ln \left( \prod_{m=1}^{M} \left( \frac{p(x_n|m_m, R_m)\pi_m(\nu)}{\tilde{C}_n(m, R, \nu)} \right)^{1[z_n=m]} \right) \right] + \ln p(y_n|x_n, z_n, \Phi, \mathcal{D}) + \text{const.} \tag{70}
\]

The first term of the right-hand side of the above equation using Eqs. (33) and (41) gives the following:

\[
\mathbb{E}_{x_n|z_n} \left[ \ln \left( \prod_{m=1}^{M} \left( \frac{p(x_n|m_m, R_m)\pi_m(\nu)}{\tilde{C}_n(m, R, \nu)} \right)^{1[z_n=m]} \right) \right]
\]

\[
= \sum_{m=1}^{M} \left[ \mathbb{I}[z_n = m] \left( \mathbb{E}_{x_n}[\pi_m(\nu)] + \mathbb{E}_{x_n|z_n}[\ln p(x_n|m_m, R_m)] - \ln (\tilde{C}_n(m, R, \nu)) \right) \right]
\]

\[
= \sum_{m=1}^{M} \left[ \mathbb{I}[z_n = m] \left( \mathbb{E}_{x_n}[\ln \nu_m] + \sum_{j=1}^{n-1} \mathbb{E}_{x_j}[\ln(1 - \nu_j)] + \frac{1}{2} \mathbb{E}_{x_n, R_m} \left[ (x_n - m_m)^T R_m (x_n - m_m) \right] - \ln (\tilde{C}_n(m, R, \nu)) \right) \right] \tag{71}
\]

where \( \mathbb{E}_{x_n, R_m} \left[ (x_n - m_m)^T R_m (x_n - m_m) \right] \) is calculated as [31]:

\[
\mathbb{E}_{x_n, R_m} \left[ (x_n - m_m)^T R_m (x_n - m_m) \right] = \nu_m (x_n - u_m)^T W_m (x_n - u_m). \tag{72}
\]

Here \( u_m, \nu_m, \) and \( W_m \) are given by Eqs. (63) and (67). Note here that even though \( \tilde{C}_n(m, R, \nu) \) is taken as a constant, it still contributes to the update of \( q(z) \).

For computing the second term of Eq. (70), we consider the predictive distribution
with $B$ and $\Sigma$ integrated out (Eq. (25)):

\[
p(y_n|x_n, z_n = m, \theta_m, D_m) = T_q(\mu^*(x_n), c^*(x_n, x_n; \theta_m)\Sigma_m; n_m - p),
\]

\[
= c_m|A_m|^{-1/2} [1_q + \frac{1}{n_m - p} (y_n - \mu^*(x_n))^T A_m^{-1} (y_n - \mu^*(x_n))]^{(n_m - p + q)/2},
\]

where we defined $c_m = (\pi(n_m - p))^{-q/2} |\Sigma_m|^{1/2}$ and $A_m = c^*(x_n, x_n; \theta_m)\Sigma_m$.

We can now compute the third term of the right-hand side of Eq. (70) as:

\[
\mathbb{E}_{q,\Theta, D}[\ln p(y_n|x_n, z_n, \Theta, D)]
= \sum_{m=1}^{M} \{ 1[z_n = m] \mathbb{E}_{q,z_m}[\ln p(y_n|z_n = m, x_n, \theta_m, D_m)] \}
\]

\[
= \sum_{m=1}^{M} \{ 1[z_n = m] \left( \ln c_m - \frac{1}{2} \mathbb{E}_{\theta_m} [\ln |A_m|] 
- \frac{n_m - p + q}{2} \mathbb{E}_{\theta_m} \left[ \ln \left[ 1_q + \frac{1}{n_m - p} (y_n - \mu^*(x_n))^T A_m^{-1} (y_n - \mu^*(x_n)) \right] \right] \right) \}.
\]

**Remark 3.** The terms above that involve the expectation, $\mathbb{E}_{\theta_m}[-]$, are computed as follows. We first approximate $q(\theta_m)$ by a Gaussian mixture as

\[
q(\theta_m) = \frac{1}{\sum_{l=1}^{L} \omega_l^2} \sum_{l=1}^{L} \omega_l^2 N(\theta_m; \mathbf{m}_l, \sigma_l^2 I_{d+1}).
\]

Let us denote the function inside of the expectation operator as $h(\theta_m)$. We can approximate $h(\theta_m)$ around each center of the Gaussian mixture $\mathbf{m}_l$ using a first-order Taylor series expansion as follows [30]:

\[
h(\theta_m) \approx \hat{h}(\theta_m) = h(\mathbf{m}_l) + \nabla h(\mathbf{m}_l)(\theta_m - \mathbf{m}_l).
\]

Then $\mathbb{E}_{\theta_m}[h(\theta_m)]$ can be approximated as

\[
\mathbb{E}_{\theta_m}[h(\theta_m)] \approx \frac{1}{\sum_{l=1}^{L} \omega_l^2} \sum_{l=1}^{L} \int_{\theta_m} \omega_l^2 N(\theta_m; \mathbf{m}_l, \sigma_l^2 I_{d+1}) \hat{h}(\theta_m) d\theta_m
\]

\[
= \frac{1}{\sum_{l=1}^{L} \omega_l^2} \sum_{l=1}^{L} \omega_l^2 h(\mathbf{m}_l).
\]

Note that the first derivative of $h(\theta_m)$, $\nabla h(\theta_m)$ does not enter this calculation. However, if a second-order Taylor expansion with respect to $\theta_m$ is used instead of Eq. (75), one would need to compute $\nabla^2 h(\mathbf{m}_l)$. 

23
Combining the terms in Eqs. (71) and (73), the variational update for $z_n$ is given by

$$q(z_n) = \prod_{m=1}^{M} \hat{p}_{n,m} [z_n = m].$$  

(76)

Here, $\hat{\rho}_{n,m} = \left( \frac{\rho_{n,m}}{\sum_{j=1}^{M} \rho_{n,j}} \right)$ is the normalized responsibility, and $\rho_{n,m}$ is given as:

$$\rho_{n,m} = \mathbb{E}_{\nu_m} [\ln \nu_m] + \sum_{j=1}^{M-1} \mathbb{E}_{\nu_j} [\ln (1 - \nu_j)] + \frac{1}{2} \mathbb{E}_{\nu_m} [\mathbf{R}_m] - \frac{d}{2} \ln(2\pi)$$

$$- \frac{1}{2} \mathbb{E}_{\nu_m \mathbf{R}_m} \left[ (\mathbf{x}_n - \mathbf{m}_m)^T \mathbf{R}_m (\mathbf{x}_n - \mathbf{m}_m) \right] - \ln \left( \mathbb{C}_m (\mathbf{\hat{m}}, \mathbf{\hat{R}}, \mathbf{\nu}) \right) + \ln \mathbb{C}_m - \frac{1}{2} \mathbb{E}_{\nu_m} [\ln |\mathbf{A}_m|]$$

$$- \frac{n_m - p + q}{2} \mathbb{E}_{\theta_m} \left[ \ln \frac{1}{n_m - p} (\mathbf{y}_n - \mathbf{\mu}_m^*(\mathbf{x}_n))^T \mathbf{A}_m^{-1} (\mathbf{y}_n - \mathbf{\mu}_m^*(\mathbf{x}_n)) \right].$$

Remark 4. Upon convergence of the VI algorithm and computation of the responsibilities $\hat{\rho}_{n,m}$, the classification of the observation data $\mathcal{D}$ is based on the maximum responsibilities. For example, we assign $(\mathbf{x}_n, \mathbf{y}_n)$ to the $i$-th model if $\arg\max_{m} \hat{\rho}_{n,m} = i$, i.e., each mixture component is taken to completely model a subset of the observation data. The clustering process is similar to the works in [24] and [27] and simplifies the training of the algorithm by having each model explaining only a small portion of the data set.

Remark 5. The number of data points assigned to each model has to be larger than $p + q$ in order to ensure a proper posterior distribution for the covariance matrix $\Sigma_m$ (see also Section 2.4.5). Therefore, any model that is assigned less than $p + q$ points is removed and the corresponding data are re-assigned to the models with the next greatest responsibility in explaining them.

2.4.5. Update of $q(\mathbf{B}_m, \Sigma_m)$

Once our data $\mathcal{D}$ are classified using $q(z)$, we proceed to update independently the hyper-parameters of each model. Let us consider the $m$-th model as an example. We first discuss the update of $q(\mathbf{B}_m, \Sigma_m)$, and then proceed to the update of $q(\theta_m)$.

From the results of Appendix A (Eq. (115)), we can directly write the update of $q(\mathbf{B}_m|\Sigma_m)$ as:

$$q(\mathbf{B}_m|\Sigma_m) = \mathcal{N}_{p \times q} \left( \mathbf{B}_m|\mathbf{B}_m, \mathbb{E}_{\theta_m} \left[ (\mathbf{H}_m^T \mathbf{A}_m^{-1} \mathbf{H}_m)^{-1} \right], \Sigma_m \right),$$  

(77)
where from Eq. (26)

$$
\mathbf{B}_m = \mathbb{E}_{\theta_m} \left[ \left( \mathbf{H}_m^T \mathbf{A}_m^{-1} \mathbf{H}_m \right)^{-1} \mathbf{H}_m^T \mathbf{A}_m^{-1} \mathbf{Y}_m \right].
$$

Similarly, from Appendix A (Eq. (117)), we can write the update of $q(\Sigma_m)$ as:

$$
q(\Sigma_m) = W_q^{-1}(\Sigma_m(n_m - p)\bar{\Sigma}_m, n_m - p),
$$

where from Eq. (118), we define:

$$
\bar{\Sigma}_m = \frac{1}{n_m - p} \mathbb{E}_{\theta_m} \left[ (\mathbf{Y}_m - \mathbf{H}_m \mathbf{B}_m)^T \mathbf{A}_m^{-1} (\mathbf{Y}_m - \mathbf{H}_m \mathbf{B}_m) \right].
$$

The expectation values are approximated here using the first-order Taylor series approximation discussed earlier in Remark [3].

**Remark 6.** The updates above require the inversion of the local covariance matrices. This scales as the cube of the number of data points assigned to a cluster. For a single MGP, the scaling is $O(N^3)$ where $N$ is the size of the data set. For a $M$-mixture MGP model, the average scaling is $O(Mm^3)$ where $m$ is the average number of points per cluster. Assuming that $m \sim N/M$, one obtains a scaling of $O(N^3/M^2)$ which is much better than $O(N^3)$ (with one cluster). Thus the mixture model scales much better than using a single MGP. In addition, these calculations can be done in parallel and fast especially if the separable covariance function approach in [5] is exploited.

### 2.4.6. Update of $q(\theta_m)$

The update of $q(\theta_m)$ is slightly more complex. This is due to the fact that the target posterior $p(\theta_m | \mathbf{B}_m, \Sigma_m, D_m)$ and thus $q(\theta_m)$ cannot be represented by any of the standard distributions (see Appendix A). A recently developed nonparametric variational inference algorithm [30] provides us a nonparametric approach for approximating $q(\theta_m)$ via Gaussian mixtures.

Let us denote the $m$-th data subset as $D_m$. The lower bound of the local evidence $\ln p(D_m)$ can be expressed as:

$$
\mathcal{L}[q] = \mathbb{E}_q \left[ \ln \frac{p(\theta_m, D_m)}{q(\theta_m)} \right] = \mathcal{H}[q] + \mathbb{E}_q[g(\theta_m)],
$$

where from Eq. (26)
where $\mathcal{H}[q]$ is the entropy of $q(\theta_m)$,

$$
\begin{align*}
\mathcal{H}[q] &= \mathbb{E}_{q(\theta_m)} \left[ \ln \pi(\theta_m | \gamma) + \ln p\left(S_m(Y; z) | S_m(X; z), \phi_m\right) \right] + \text{const} \\
&= \ln \pi(\theta_m | \gamma) + \mathbb{E}_{q(\theta_m)} \left[ \ln p\left(S_m(Y; z) | S_m(X; z), \theta_m, B_m, \Sigma_m\right) \right] + \text{const}, \\
&= \sum_{j=1}^{L} \omega_j^2 \mathcal{H}[q_j],
\end{align*}
$$

and using Eq. (15) (also Eq. (113)):

$$
\begin{align*}
\mathbb{E}_{q(\theta_m)} \left[ \ln p\left(S_m(Y; z) | S_m(X; z), \theta_m, B_m, \Sigma_m\right) \right] &= \mathbb{E}_{q(\theta_m)} \left[ \ln \exp \left( \frac{1}{2} \text{tr} \left( \Sigma_m^{-1} (S_m(Y; z) - H_m B_m)^T A_m^{-1} (S_m(Y; z) - H_m B_m) \right) \right) \right] \nonumber \\
&= -\frac{1}{2} \text{tr} \left[ \mathbb{E}[\Sigma_m^{-1}](S_m(Y; z) - H_m \mathbb{E}[B_m])^T A_m^{-1} (S_m(Y; z) - H_m \mathbb{E}[B_m]) \right] \\
&= -\ln \left( 2\pi e n^d / |A_m|^{n/2} \right) \\
&= -\frac{1}{2} \text{tr} \left( \Sigma_m^{-1} \right) + \text{const}. \\
&= -\frac{1}{2} \sum_{i=1}^{n} \sigma_i^2 \ln q_i, \\
&= -\frac{1}{2} \sum_{i=1}^{n} \sigma_i^2 \ln q_i + \text{const}, \\
&= -\frac{1}{2} \sum_{i=1}^{n} \sigma_i^2 \ln q_i + \text{const}, \\
&= -\frac{1}{2} \sum_{i=1}^{n} \sigma_i^2 \ln q_i + \text{const}, \\
&= -\frac{1}{2} \sum_{i=1}^{n} \sigma_i^2 \ln q_i + \text{const}, \\
&= -\frac{1}{2} \sum_{i=1}^{n} \sigma_i^2 \ln q_i + \text{const},
\end{align*}
$$

Following [30] (see also Remark 3), we choose the distribution of $q(\theta_m)$ to be a weighted Gaussian mixture with isotropic covariance as in Eq. (74).

Next, we compute the lower bound of the entropy term $\mathcal{H}[q]$, and then maximize the new lower bound of the local evidence. The lower bound of the entropy can be found using Jensen’s inequality [40].

$$
\mathcal{H}[q] = -\int_{\theta_m} q(\theta_m) \ln q(\theta_m) d\theta_m \\
= -\int_{\theta_m} \frac{1}{\Sigma_{l=1}^{L} \omega_l^2} \sum_{i=1}^{L} \omega_i^2 \mathcal{H}[q_i] + \text{const} \\
= -\frac{1}{\Sigma_{l=1}^{L} \omega_l^2} \sum_{i=1}^{L} \omega_i^2 \int_{\theta_m} \mathcal{H}[q_i] + \text{const} \\
\geq -\frac{1}{\Sigma_{l=1}^{L} \omega_l^2} \sum_{i=1}^{L} \omega_i^2 \ln q_i, \\
$$

where $q_i = \sum_{j=1}^{L} \omega_j^2 q_{ij}, q_{ij} = \mathcal{N}(m_i; m_i, (\sigma_i^2 + \sigma_j^2) I_{d+1})$. The proof can be found in Appendix D.

We now look at the expected log local joint $g(\theta_m)$,

$$
\mathbb{E}_{q_m}[g(\theta_m)] = \frac{1}{\Sigma_{l=1}^{L} \omega_l^2} \sum_{i=1}^{L} \omega_i^2 \int_{\theta_m} \mathcal{H}[\theta_m; m_i, \sigma_i^2 I_{d+1}] g(\theta_m) d\theta_m.
$$
The local joint $g(\theta_m)$ can be approximated around each site of the proposals $m_l$ with a second-order Taylor series expansion,

$$g(\theta_m) \approx \hat{g}_l(\theta_m) = g(m_l) + \nabla g(m_l)(\theta_m - m_l) + \frac{1}{2}(\theta_m - m_l)^T H_l(\theta_m - m_l), \quad (85)$$

where $H_l = \nabla^2_{\theta_m} g(\theta_m)$ is the Hessian matrix. The approximate expectation can be then written as:

$$E[q[g(\theta_m)]] = \frac{1}{\sum_{l=1}^L \omega_l^2} \sum_{j=1}^L \omega_l^2 \mathcal{N}(\theta_m; m_l, \sigma_l^2) \hat{g}_l(\theta_m)$$

$$= \frac{1}{\sum_{l=1}^L \omega_l^2} \sum_{j=1}^L \omega_l^2 \left\{g(m_l) + \frac{\sigma_l^2}{2} tr(H_l)\right\}. \quad (86)$$

The proof can be found in Appendix E. Finally, combined with the bound of the entropy term in Eq. (83), we derive the following approximation of the lower bound of the local evidence,

$$L_2[q(\theta)] = \frac{1}{\sum_{l=1}^L \omega_l^2} \sum_{j=1}^L \omega_l^2 \left\{g(m_l) + \frac{\sigma_l^2}{2} tr(H_l)\right\} - \frac{1}{(\sum_{l=1}^L \omega_l^2)^2} \sum_{l=1}^L \omega_l^2 \ln q_l. \quad (87)$$

The variational parameters $\omega_l$, $m_l$ and $\sigma_l$ are learnt by a gradient ascent method. Following [30], in the update of $m_l$, we only use the first-order approximated lower bound of the local evidence to avoid the calculation of the gradient of the Hessian trace $tr(H_l)$. The first-order approximation is given by

$$L_1[q(\theta)] = \frac{1}{\sum_{l=1}^L \omega_l^2} \sum_{j=1}^L \omega_l^2 g(m_l) - \frac{1}{(\sum_{l=1}^L \omega_l^2)^2} \sum_{l=1}^L \omega_l^2 \ln q_l. \quad (88)$$

The nonparametric variational inference procedure is summarized in Algorithm 1 (\eta_1, \eta_2, \eta_3 appropriate learning rates). The final forms of the needed derivatives $\frac{\partial L_1[q]}{\partial m_l}$, $\frac{\partial L_1[q]}{\partial \sigma_l}$ and $\frac{\partial L_1[q]}{\partial \omega_l}$ are given in Appendix F.

Finally, with $\varsigma_1, \varsigma_2$ given tolerances, we summarize the overall variational inference approach in Algorithm 2.

2.5. Application to uncertainty quantification

As discussed Section 1, the problem of UP is to propagate the uncertainty of the random variables $\xi$ through $f(\cdot)$ and characterize the statistics of the response such
Algorithm 1 Nonparametric variational inference for MGP model

Input: training data set $D_m$, number of kernels $L$.

Initialize: $(\omega_l, m_l, \sigma_l)_{l=1}^L$ randomly. Set $t = 0$.

repeat
  for $l = 1, \ldots, L$ do
    $(m_l)^{(t+1)} = (m_l)^{(t)} + \eta_1 \frac{\partial L_1[q]}{\partial m_l}$.
  end for

  for $l = 1, \ldots, L$ do
    $(\sigma_l)^{(t+1)} = (\sigma_l)^{(t)} + \eta_2 \frac{\partial L_2[q]}{\partial \sigma_l}$.
  end for

  for $l = 1, \ldots, L$ do
    $(\omega_l)^{(t+1)} = (\omega_l)^{(t)} + \eta_3 \frac{\partial L_3[q]}{\partial \omega_l}$.
  end for

  set $t = t + 1$.

until change of $L_2[q(\theta_m)]$ is less than $\delta$.

as the mean (see Eq. (4)), the covariance (see Eq. (5)), the PDF of the output at a particular location and time (see Eq. (6)), and so on. We exploit the Bayesian nature of the surrogate we build in order to derive error bars for any computed statistic. These error bars correspond to the epistemic uncertainty induced by the limited amount of data in $D$ which is, in turn, captured by the approximate posterior over the parameters $q(\Psi)$ of Eq. (57). The key idea is that every sample from $q(\Psi)$ yields a sample response surface $\tilde{f}(\cdot; \Psi, D)$ which may be interrogated for the statistics. Mathematically, let $Q[\cdot]$ be a statistic of interest, i.e. a functional of the response surface. Then, we can obtain a probabilistic estimate for this statistic as follows:

$$p(Q|D) = \int q(\Psi) \delta \left( Q - Q \right) d\Psi. \quad (89)$$

The uncertainty in $p(Q|D)$ is due to the limited number of observations contained in $D$. 

28
Algorithm 2 Computing the approximation $q(\Psi)$ to the posterior $p(\Psi|D, \gamma, \alpha_0)$

Given: $D = [X, Y], \alpha_0, \gamma_r, \gamma_e$ and tolerances $\varsigma_1, \varsigma_2, \text{and } \delta$.

Set $t_1 = 0$ and the truncation level $M$

Initialize $q(\Psi)$ using the prior distribution

Randomly set $q^0(Z)$ by sampling from the prior distribution of the DP using Eq. (41)

while $|L^{t+1}(q, D) - L^t(q, D)| < \varsigma_1$ do

Cluster the observations based on $q^t(Z)$ as discussed in Section 2.4.4. Let $M_r$ be the number of remaining clusters.

for $m = 1, \ldots, M_r$ do

Update of $q^{t+1}(m_m)$ using Eq. (63)

Update of $q^{t+1}(R_m)$ using Eq. (67)

Set $t_2 = 0$

while $|L^{t_2+1}(q_m, D_m) - L^{t_2}(q_m, D_m)| < \varsigma_2$ do

Update of $q^{t_2+1}(B_m|m_m)$ using Eq. (77)

Update of $q^{t_2+1}(\Sigma_m)$ using Eq. (79)

Update $q^{t_2+1}(\theta_m)$ using Algorithm 1 in Section 2.4.6.

Set $t_2 = t_2 + 1$.

end while

end for

Update $q^{t+1}(\nu)$ by maximizing $L^t(q(\nu), D)$ using Eq. (60).

Update $q^{t+1}(z)$ by maximizing $L^t(q(z), D)$ using Eq. (76).

Calculate $\hat{C}_n(m, \hat{R}, \hat{\nu})$ in Eq. (56) for all observations.

Set $t_1 = t_1 + 1$.

end while
2.5.1. The predictive distribution

In general, we want to predict on a denser spatial points \( S' \in \mathbb{R}^{n \times d} \) and/or time steps \( t' \in \mathbb{R}^{n} \). Let \( \xi' \) be one sample from the known distribution \( p(\xi), s' \in S', \) and \( t' \in t' \), and we denote \( y' \) as the corresponding unknown output at the point \( x' = (\xi', s', t') \).

Then, the joint distribution of \( y' \) and the hidden variables \( z' \) given the new input \( x' \), all the observations and the parameters \( \Psi \) can be written as:

\[
p(y', z'|x', \Psi, D) = p(y'|x', z', \Phi, D)p(z'|x', m, R, \nu) = p(y'|x', S_c(X; z), S_c(Y; z), \phi_c)p(z'|x', m, R, \nu).
\]

The first term of the right hand-side of the above equation is the predictive distribution of the \( z' \)-th latent MGP given by Eq. (20). The second term represents the predictive responsibility that is given by Eq. (42):

\[
p(z'|x', m, R, \nu) = \frac{p(x'|z', m_c, R_c)p(z'|\nu)}{\sum_{z'} p(x'|z', m_c, R_c)p(z'|\nu)},
\]

where \( p(x'|z', m_c, R_c) \) is given by Eq. (43), \( p(z'|\nu) \) is given by Eq. (41).

We may first multiply the joint distribution with the approximate posteriors \( q(B_m, \Sigma_m), q(m), q(R), \) and \( q(\nu) \), and integrate them out. We have:

\[
p(y', z'|x', z, \Theta, D) = \int p(y'|x', z', \Phi, D)p(z'|x', m, R, \nu)q(B_c, \Sigma_c)q(m)q(R)q(\nu)dm dR d\nu.
\]

\[
= \int p(y'|x', z', \Phi, D)q(B_c, \Sigma_c)dB_c d\Sigma_c \int p(z'|x', m, R, \nu)q(m)q(R)q(\nu)dm dR d\nu.
\]

Here recall \( \Theta = \{\theta_1, \ldots, \theta_M\} \).

The first integral term can be calculated as

\[
p(y'|x', S_c(X; z), S_c(Y; z), \theta_c) = \int p(y'|x', S_c(X; z), S_c(Y; z), \phi_c)q(B_c, \Sigma_c)dB_c d\Sigma_c
\]

\[
= p(y'|x', S_c(X; z), S_c(Y; z), \theta_c, \bar{B}_c, \bar{\Sigma}_c),
\]

where \( \bar{B}_c \) and \( \bar{\Sigma}_c \) are the posterior mean given by Eq. (77) and Eq. (79), respectively, and they depend on \( \theta_c \).

The second integral term is analytically intractable, so we simply approximate it using the mean of \( q(m), q(R) \) and \( q(\nu) \). Mathematically, we take \( q(m) = \delta(m - \bar{m}) \),

30
where $\bar{m}$ is the posterior mean of $m$. For $R$ and $\nu$, we use a similar trick. Then, we can write
\[
p(z^*|x^*) = \int \frac{p(x^*|z^*, \bar{m}_x, \bar{R}_x) p(z^*|\nu)}{\sum_{z^*} p(x^*|z^*, \bar{m}_x, \bar{R}_x) p(z^*|\nu)} q(m) q(\nu) dm dR d\nu
\]
\[
\approx \frac{p(x^*|z^*, \bar{m}_x, \bar{R}_x) p(z^*|\nu)}{\sum_{z^*} p(x^*|z^*, \bar{m}_x, \bar{R}_x) p(z^*|\nu)},
\]

Now the joint distribution $p(y^*, z^*|x^*, \Psi, D)$ can be simplified as
\[
p(y^*, z^*|x^*, z, \Theta, D) = p(y^*|x^*, S_c(X; z), S_c(Y; z), \theta_c) p(z^*|x^*).
\]

We can further multiply $q(z)$ and integrate $z$ out. Here, as discussed in Remark 4, we choose $\tilde{z}$ to be the MAP estimate of $z$ as
\[
\tilde{z} = \arg \max_z q(z). \tag{90}
\]
Then, we take $q(z) \approx \delta(\tilde{z} - z)$. Making use of this approximation, we have:
\[
p(y^*, z^*|x^*, \Theta, D) = p(y^*|x^*, S_c(X; \tilde{z}), S_c(Y; \tilde{z}), \theta_c) p(z^*|x^*).
\]

Finally, we may integrate out $z^*$ to obtain:
\[
p(y^*|x^*, \Theta, D) \approx \sum_{z^*} p(y^*, z^*|x^*, \Theta, D)
\]
\[
= \sum_{z^*} p(y^*|x^*, S_c(X; \tilde{z}), S_c(Y; \tilde{z}), \theta_c) p(z^*|x^*).
\]

From Eq. (23), the predictive distribution $p(y^*|x^*, S_m(X; \tilde{z}), S_m(Y; \tilde{z}), \theta_m)$ follows a $q$-dimensional student-$T$ distribution:
\[
p(y^*|x^*, S_m(X; \tilde{z}), S_m(Y; \tilde{z}), \theta_m) = T_q \left( y^* \left| \mathbf M^{(m)}(x^*, \theta_m), \mathbf C^{(m)}(x^*, \theta_m); n_m - p \right. \right), \tag{91}
\]
with the mean and covariance functions given:
\[
\mathbf M^{(m)}(x^*, \theta_m) = \mathbf h(x^*) \mathbf B_m + \mathbf a_m(x^*)^T \mathbf A_m^{-1} (\mathbf Y_m - \mathbf H_m \mathbf \bar{B}_m),
\]
\[
\mathbf C^{(m)}(x^*, \theta_m) = c^{**}(x^*, \mathbf \Sigma_m).
\]

with $\mathbf B_m$, $\mathbf \Sigma_m$, and $c^{**}(\cdot)$ defined in Eq. (23). The predictive mean function $\mathbf M^{(m)}(x^*, \theta_m)$ is thought of as a sampled surrogate model. It is subsequently used to calculate the statistics of interest.
A graphical illustration of calculating \( p(y^*|x^*, \Theta, D) \) is given in Fig. 2.

Figure 2: Graphical model illustration of doing predictions using the proposed framework. \( f^{(m)} \) and \( \theta_m \) are known to us from the variational inference algorithm discussed above. \( B_m, \Sigma_m, \mu_m, \) and \( \nu_m \) are integrated out from the framework. \( x^* \) denotes the new input, \( z^* \) gives the predictive responsibilities for each mixture component, and then \( y^* \) is calculated as a weighted combination of the predictions given by each mixture component, as in Eq. (91).

### 2.5.2. Analytic first- and second-order statistics

For one sample \( \Theta' \), the first and second-order statistics of interest can then be evaluated analytically using the sampled surrogate model \( M'^{(m)}(x^*, \theta_m') \). From now on, we ignore the dependence on \( \theta_m' \) in order to lessen the notational burden. The mean at the spatial locations \( s^* \) and time \( t^* \) is:

\[
M^*(s^*, t^*) := \sum_{m=1}^{M} \int M'^{(m)}(x^*) p(z^* = m|x^*) p(\xi^*) d\xi^*. \tag{92}
\]

Recall that \( x^* = (\xi^*, s^*, t^*) \) and the integration is over only the stochastic inputs \( \xi^* \). This can also be written as:

\[
M^*(s^*, t^*) = \sum_{m=1}^{M} \left[ \zeta^{(m)}_h \bar{B}_m + \zeta^{(m)}_a A_m^{-1} (Y_m - H_m \tilde{B}_m) \right], \tag{93}
\]

where

\[
\tilde{\zeta}^{(m)}_h = \int h(x^*) p(z^* = m|x^*) p(\xi^*) d\xi^*,
\]

\[
\tilde{\zeta}^{(m)}_a = \int a_m(x^*) p(z^* = m|x^*) p(\xi^*) d\xi^*.
\]

32
Now let $i, j \in \{1, \ldots, q\}$ be two arbitrary outputs. The covariance matrix between all spatial and time test points is defined by:

$$C_{ij}(s^*, t^*) = \int (Y_i - M_i^*)(Y_j - M_j^*)^T p(\xi^*) d\xi^*,$$

$$= \int \left( \sum_{m=1}^{M} \beta_m \beta_r m_i^{(m)}(x^*) (\xi^* = m|x^*) - M_i^* \right) \left( \sum_{r=1}^{M} \beta_r \beta_m m_j^{(m)}(x^*) (\xi^* = r|x^*) - M_j^* \right)^T p(\xi^*) d\xi^*,$$

$$= \int \left( \sum_{m=1}^{M} \sum_{r=1}^{M} \beta_m \beta_r m_i^{(m)}(x^*) m_j^{(m)}(x^*) \right) p(\xi^*) d\xi^* - M_i^* (M_j^*)^T,$$  \hfill (94)

where $\beta_m = p(x^* = m|x^*)$ and $\beta_r = p(x^* = r|x^*)$, and the subscripts $i$ and $j$ indicate columns of the associated matrices. This covariance matrix can be evaluated by:

$$C_{ij}(s^*, t^*) = \sum_{m=1}^{M} \sum_{r=1}^{M} [\kappa_{hh}^{mr} + \kappa_{ha}^{mr} + \kappa_{aa}^{mr}] - M_i^* (M_j^*)^T,$$  \hfill (95)

where

$$\kappa_{hh}^{mr} = \int \beta_m \beta_r (h(x^*) B_m) (\tilde{B}^T h(x^*)^T) p(\xi^*) d\xi^*,$$

$$\kappa_{ha}^{mr} = \int \beta_m \beta_r (h(x^*) B_m) (\tilde{Y}^T A^{-1} a_r(x^*)^T) p(\xi^*) d\xi^*,$$

$$\kappa_{aa}^{mr} = \int \beta_m \beta_r (a_m(x^*)^T A^{-1} \tilde{Y}_m) (\tilde{Y}^T A^{-1} a_r(x^*)^T) p(\xi^*) d\xi^*,$$

where $\tilde{Y}_m = Y_m - H_m \tilde{B}_m$ and $\kappa_{aa}^{mr} = (\kappa_{ha}^{mr})^T$.

Higher-order statistics can be obtained via Monte Carlo method. By gathering all the sampled response surfaces, one can obtain a probabilistic measure of any statistics of interest with not only mean predictions but also additional error bars. The whole algorithm is summarized in Algorithm 3.

**Algorithm 3** Sampling the posterior of statistics of interest.

**Given:** Observed data $D$, and approximated posterior $q(\Theta)$, 

**Objective:** Repeatedly sample from the posterior of statistics

1. Sample $\Theta'$, from the approximated posterior distribution
2. Sample a response surface and calculate the first and second-order statistics
3. Interrogate the obtained response surface (analytically or via MC)
3. Numerical Examples

In this section, we first study the benchmark Kraichnan-Orszag (KO) problem to demonstrate the unique features of the proposed framework. Then, we continue with the study of a heterogeneous oil reservoir problem. All examples considered are run on massively parallel computers operated by the National Energy Scientific Computing Center (NERSCC) [41]. The tolerance $\delta$ of Algorithm 1 is taken as $\delta = 10^{-4}$. The tolerances of Algorithm 2 are $\varsigma_1 = 10^{-3}$, and $\varsigma_2 = 10^{-3}$. The variational inference algorithm converges in no more than 1,000 iterations.

3.1 Kraichnan-Orszag problem

The transformed Kraichnan-Orszag three-mode problem is described by the following dynamical system [12, 42]

$$
\begin{align*}
\frac{dy_1}{dt} &= y_1 y_3, \\
\frac{dy_2}{dt} &= -y_2 y_3, \\
\frac{dy_3}{dt} &= -y_1^2 + y_2^2,
\end{align*}
$$

subject to random initial conditions at $t = 0$. A discontinuity in the solution occurs when $y_1(0)$ and $y_2(0)$ cross the plane $y_1 = 0$ and $y_2 = 0$. The deterministic solver we use is a 4-th order Runge-Kutta method as implemented in the GNU Scientific Library [41].

3.1.1 One-dimensional case

Let us first consider the one-dimensional case with initial conditions as

$$
\begin{align*}
y_1(0) &= 1, \\
y_2(0) &= 0.1 \xi, \\
y_3(0) &= 0,
\end{align*}
$$

where $\xi \sim \mathcal{U}([-1, 1])$. This problem has a discontinuity at $\xi = 0$.

The stochastic input $\xi$, plus the time variable $t$, determine the input dimension of our model as $d = d_\xi + d_t = 2$. The responses are recorded at 40 equidistant time
steps in between the time interval $[0, 10]$, i.e., $n_t = 40$. The hyper-parameters are selected as $\alpha = 1, \gamma_r = 5, \gamma_\epsilon = 10^{-6}$, $u_0 = 0$, $R_0 = 10^{-3}I_d$, $W_0 = 10^{-3}I_d$ and $v_0 = d$. The truncation level $M$ is set to 200. We train the model with $n_\xi = 51$ and $n_\xi = 98$ observations. The observations are initially randomly classified. The variational inference algorithm gradually obtains the optimal clustering by maximizing the lower bound of the evidence. Figure 3(a) shows the initial random clustering of the input data. Figure 3(b) shows the computed clustering at an intermediate stage of the algorithm. It can be seen that this clustering is not optimized because there are clusters formed with data across the discontinuity ($\xi = 0$). Recall that these clusters not only decompose the stochastic space but also the temporal space. The final clustering pattern obtained is shown in Fig. 3(c). Eventually twelve groups are left after running the variational inference algorithm, six on the left side of the discontinuity, and another six on the right side. Note that there is no decomposition of the time space after the algorithm converges. The convergence of the variational inference algorithm can also be seen by plotting the number of clusters at each iteration step (Fig. 4). From this figure, we can see that increasing the size of the training data set increases the number of the mixture components. Finally, note that the number of clusters decreases with iteration with some minor oscillation that is possibly due to the local adjustments of the constructed model.

Figure 3: KO1 - demonstration of the evolution of the clustering with $n_\xi = 51$ observations: (a) the initial clustering; (b) the clustering at an intermediate iteration of the variational inference algorithm; (c) the final clustering. The number of iterations and the number of clusters selected are shown above each figure.
To show the convergence of the posterior distribution, $q(z)$ (Eq. (76)), we randomly pick two observations at input points, $(-0.01, 0.8)$ and $(0.9, 6)$, and plot the corresponding normalized responsibilities with respect to all the mixture components (in this example, 200) at each iteration step, as in Fig. 5. At the early iterations and for the point $(-0.01, 0.8)$ very close to the discontinuity, note that a number of clusters are taking responsibility in explaining it but eventually only one dominant cluster is selected. For the point $(0.9, 6)$ close to the boundary, one dominant cluster is selected from the very beginning of the algorithm. This implicitly shows that the approximation of the normalization term $\hat{C}_m(\hat{m}, \hat{R}, \hat{\nu})$ in Eq. (76) does not affect the convergence of the variational inference algorithm. The convergence of the lower bound of the model evidence is given in Fig. 6.
Figure 5: KOI - convergence of the responsibility for \( n_{\xi} = 51 \) observations: (a) for input point \((-0.01, 0.8)\); (b) for input point \((0.9, 6)\). We start with \( M = 200 \) components, and as the variational inference algorithm proceeds most of the components vanish and therefore do not contribute to the responsibility of the corresponding component for a query point. The algorithm eventually provides one dominant component for each query point.

Figure 6: KOI - convergence of the lower bound for the model evidence.

The variational inference algorithm converges fast because the update equations for the proposed posterior distributions \( q(\nu) \) (Eq. (60)) and \( q(z) \) (Eq. (76)) are analytical. The update of \( q(\theta) \) is a bit more complex but it does not deteriorate the performance of the variational inference algorithm due to the nature of the gradient descent algorithm.

From the variational inference algorithm, the approximated posterior distribution
of parameters of interest can be obtained. Then we draw 100 samples from the approximated posterior distribution and we make predictions of the statistics of interest. For each parameter sample, the predictions are made at 100 equidistant time steps \( (n^* = 100) \) in \([0, 10]\). The statistics of interest can be calculated semi-analytically, using Eqs. (93) and (95). We then calculate the mean of the statistics as well as the 95\% confidence intervals (error bars). With \( n_\xi = 98 \) observations, the constructed model can capture the mean and also the variance of each output quite accurately, as shown in Fig. 7.

![Figure 7: KO1 - The blue curve is the mean of the statistic of interest (mean and variance) predicted by the model while the gray area shows 95\% confidence intervals. The red curve is the MC result using \( 10^6 \) samples. The first row provides the predictive means for \( y_1, y_2 \) and \( y_3 \) with \( n_\xi = 98 \). The second row shows the corresponding predictive variances.](image)

We further evaluate the probability density functions (PDFs) of the responses at certain time steps. The predictive PDFs are obtained in a similar way as discussed above: (1) we sample 100 parameters from the approximated posteriors; (2) for each set of the parameters, we draw 50,000 samples from \( p(\xi) \); (3) we predict the response for each of these \( \xi \); and (4) we use kernel density estimator to approximate the desired
PDFs. The mean of the predicted PDFs with \( n_\xi = 98 \) for each output at various time locations as well as the error bars are compared to the MC results obtained with \( 10^6 \) samples as shown in Fig. 8.

Figure 8: KOI - Comparison of the predicted PDFs with \( n_\xi = 98 \) to the PDFs obtained with MC. Each row depicts the PDFs of \( y_1 \), \( y_2 \) and \( y_3 \) for time \( t = 6, 8, 10 \), respectively. The blue curve is the mean of the statistic predicted by the model while the gray area shows 95% confidence intervals. The red curve is the MC estimate obtained using \( 10^6 \) samples.

In comparison to the multi-element GP approach in [16], the present method can automatically cluster the observations, and the discontinuities can be discovered without explicitly decomposing the stochastic space. Due to the nature of the variational inference algorithm, the posterior distributions of interest are more efficiently approximated than using MCMC [5]. Also by the nature of the mixture model, less data are
assigned to each GP component thus avoiding forming large size covariance matrices.

3.1.2. Two-dimensional case

Let us now consider the two-dimensional case. The initial conditions for the problem are taken as:

\[
\begin{align*}
    y_1(0) &= 0, \\
    y_2(0) &= 0.1 \xi_1, \\
    y_3(0) &= \xi_2,
\end{align*}
\]

where \( \xi_i \sim U([-1, 1]), \ i = 1, 2. \)

The input dimension for the 2D problem is \( d = d_\xi + d_t = 3. \) In this example, there is a discontinuity occurring as the initial condition cross the line \( \xi_1 = 0. \) The responses are also recorded at 40 equidistant time steps in between the time interval \([0, 10]\), i.e., \( n_t = 40. \) The hyper-parameters are selected as for the 1D case. The truncation level \( M \) is set to be 800. The model is trained with \( n_\xi = 100, 200 \) and 400 observations and the algorithm stops at 52, 85, and 112 clusters, respectively. The decomposition of the time space has been observed around the discontinuity when the algorithm converges.

Figure 9 shows the comparison of the mean predictions of the mean of each output as well as the error bars with \( n_\xi = 200 \) to the MC estimates with \( 10^6 \) samples. As shown from the figure, the mean of each output is very accurately captured using the constructed model with \( n_\xi = 200. \) The variance of each output is gradually captured with an increase of the number of observations, and the corresponding error bars are also shrinking, as shown in Fig. 10. The predicted probability density functions (PDFs) for the responses are obtained in a similar way to that described in the 1D example. Here, we only compare the predicted PDFs for \( y_2 \) at various time locations to the MC results. The mean predictions as well as the error bars for each output at various times are compared to the MC estimates, as shown in Fig. 11. From these figures, even with few numbers of observations, the discontinuity can be discovered, and as expected increasing the number of observations results in more accurate predictions.
Figure 9: KO2 - Comparison of predictive means for each output with the MC computed means for $n_\xi = 200$. The blue curve is the mean of the statistic predicted by the model while the gray area shows 95% confidence intervals. The red curve is the MC estimate obtained using $10^6$ samples.

Figure 10: KO2 - Comparison of predictive variances for each output with the MC computed means. The top row provides the results with $n_\xi = 200$, and the bottom row gives the predictions with $n_\xi = 400$. The blue curve is the mean of the statistic predicted by the model while the gray area shows 95% confidence intervals. The red curve is the MC result obtained using $10^6$ samples.
Figure 11: KO2 - Convergence plots of PDFs for $y_2$ at different time steps and different numbers of observations. The first column corresponds to $n_\xi = 100$, the second to $n_\xi = 200$, and the third to $n_\xi = 400$. Each row depicts the PDF of $y_2(t)$ at times $t = 4, 6, 8, 10$. The blue curve is the mean of the statistic predicted by the model while the gray area shows 95% confidence intervals. The red one is the MC estimates with $10^6$ samples.
3.1.3. Three-dimensional case

Finally, let us consider the more difficult and computationally demanding three-dimensional case. The initial conditions for the problem are:

\[
\begin{align*}
    y_1(0) &= \xi_1, \\
    y_2(0) &= \xi_2, \\
    y_3(0) &= \xi_3,
\end{align*}
\]

where

\[\xi_i \sim U([-1, 1]), \ i = 1, 2, 3.\]

The hyperparameters are set as in the previous cases, the truncated level \(M\) is set to 2000. The model is trained with \(n_\xi = 1000\), and \(n_\xi = 4000\) observations. For this case, the algorithm stops at 376, and 982 clusters for \(n_\xi = 1000\) and \(n_\xi = 4000\), respectively. The decomposition of the time space has also been observed at convergence. Again, the constructed models can capture the mean and variance of the statistics of interest very well. As shown in Fig. 12, the increase of the number of observations results in a better predictive performance of the constructed model. Similarly, we compare the predictive PDFs for \(y_2\) and \(y_3\) at \(t = 8, 10\) with \(n_\xi = 4000\) to the MC results with \(10^6\) samples. From Fig. 13 we can see that the predicted PDFs agree well with the MC estimates as expected.
Figure 12: KO3 - Comparison of predictive variances for \( y_1(t) \), \( y_2(t) \), and \( y_3(t) \) with \( n_\xi = 1000 \) and 4000 to the MC variances with \( n_\xi = 10^6 \). The blue curve is the mean of the predicted variance while the gray area shows 95% confidence intervals. The red curve is the MC result.
3.2. Flow through porous media

In this section, we study a heterogeneous oil reservoir flow problem in a confined domain $\Lambda$ in $\mathbb{R}^d$. A standard quarter-five spot problem with one injection well on the bottom left and one production well on the top right is considered. We consider, immiscible, and incompressible two-phase flow (water and oil) $[7, 44, 45]$. The capillary pressure and gravity are not included in the model.

From Darcy’s law, we can first write the pressure equation for the flow as:

$$\nabla \cdot v = q, \text{ where } v = -(A_o + A_w)K\nabla p,$$

where the subscripts $o, w$ represent oil phase and water phase, respectively. $v$ is the total velocity, i.e., $v = v_o + v_w$. $q = q_w + q_o$ is the source/sink term, which models the injection/production well in this problem. No-flow boundary conditions are considered. $K$ is the location-dependent permeability tensor. $p$ is the pressure, and here we take
\( p_a = p_w = p \) due to the assumption of no capillary pressure. \( \lambda_a \), with \( \alpha = o, w \), is the phase mobility given by \( \lambda_a = k_{ro}/\mu_w \), where \( k_{ro} = k_{ro}(s) \) is the relative permeability depending on the saturation \( s_\alpha \) (fraction of the void occupied by phase \( \alpha \)), and \( \mu_w \) is the viscosity of phase \( \alpha \). The relative permeability models the reduced conductivity of a phase due to the presence of other phases, and is, according to common practice, assumed to be a function of the saturation only. In this work, we use

\[
k_{rw} = (s')^2, \quad k_{ro} = (1 - s')^2, \quad s' = \frac{s - s_{wc}}{1 - s_{wc} - s_{or}},
\]

with \( s_{wc} = s_{or} = 0.2 \) and initial saturation \( s_0 = s_{wc} \). Also, \( s_{or} \) is the irreducible oil saturation, i.e., the lowest oil saturation that can be achieved by displacing oil by water, and \( s_{wc} \) is the connate water saturation, i.e., the saturation of water trapped in the pores of the rock during formation of the rock.

By introducing the concept of fractional flow \( f_\alpha = \lambda_a/\lambda \), where \( \lambda = \lambda_o + \lambda_w \) is the total mobility, we can write the phase velocity \( v_\alpha = f_\alpha v \). Note \( f_\alpha \) is also dependent on the saturation \( s_\alpha \). Then, combined with the conservation equation of mass for each phase \( \alpha \), we may write the saturation equation (fluid transport), as follows:

\[
\phi \frac{\partial s_\alpha}{\partial t} + \nabla \cdot (f_\alpha v) = q_\alpha,
\]

where \( \phi \) is porosity, which is modeled as a random field in this problem rather than a constant as in [4, 5]. The porosity has been shown to have a strong correlation with the permeability, as discussed in Section 1. Since \( s_o + s_w = 1 \), only the water saturation \( s_w \) is considered below.

The fraction of water/oil in the produced fluid as a function of time, \( F(t) \), is of great interest [4, 7], and it is known as the water-cut curve defined as

\[
F(t) = \frac{\int_{\partial \Lambda_{out}} f_w(v \cdot n)dx}{\int_{\partial \Lambda_{out}} (v \cdot n)dx},
\]

where \( \partial \Lambda_{out} \) is the outflow boundary, \( n \) is the normal to the boundary. Also note that the time, \( t \), is measured in days. This is different from the traditional learning of the water-cut curve [4], where \( t \) is considered as a dimensionless time (PVI, pore volume injected). PVI is not considered in this paper since the porosity is also modeled as
a random field (thus, for different realizations, if we considered the same amount of water injected, the corresponding PVI would be different).

3.2.1. Discretizing the pressure equation

The pressure equation is discretized with a mixed FEM method\cite{46,44}. Recall in this work, we consider no flow boundary condition, i.e., $v \cdot n = 0$ and an extra constraint, $\int_\Lambda pdx = 0$, is added to close the system. To derive the mixed formulation, we first define the Sobolev space

$$H^d_0(\Lambda) = \left\{ v \in \left( L^2(\Lambda) \right)^d \mid \nabla \cdot v \in L^2(\Lambda) \text{ and } v \cdot n = 0 \text{ on } \partial \Lambda \right\}.$$ \hspace{1cm} (101)

The mixed-finite element discretization in domain $\Lambda$ seeks a pair $(v, p) \in U \times V$, where $U$ and $V$ are finite-dimensional subspaces of $H^d_0(\Lambda)$ and $L^2(\Lambda)$, respectively, such that,

$$\int_\Lambda v \cdot (\Lambda K)^{-1} u dx - \int_\Lambda \nabla \cdot u dx = 0, \quad \text{for all } u \in U, \quad \text{(102)}$$

$$\int_\Lambda \nabla \cdot v dx = \int_\Lambda l q dx, \quad \text{for all } l \in V. \quad \text{(103)}$$

Now, let us partition the domain $\Lambda$ into mutually disjoint grid cells as $\Lambda = \{\Lambda_m\}$. The basis functions $\chi_m$ and $\psi_{ij}$ on a cell $\Lambda_m$ and edge $\gamma_{ij} = \partial \Lambda_i \cap \partial \Lambda_j$ are defined respectively by:

$$\chi_m = \begin{cases} 
1, & \text{if } x \in \Lambda_m, \\
0, & \text{elsewise,}
\end{cases} \quad \text{(104)}$$

and

$$\psi_{ij} \cdot n_{kl}|_{x_i} = \begin{cases} 
1, & \text{if } \gamma_{ij} = \gamma_{kl}, \\
0, & \text{elsewise.}
\end{cases} \quad \text{(105)}$$

Thus, we can write $p = \sum p_m \chi_m$ and $v = \sum v_{ij} \psi_{ij}$, where $v_{ij} = \int_{\gamma_{ij}} v \cdot n_{ij} dx$, where $n_{ij}$ is the unit normal to $\gamma_{ij}$ from $\Lambda_i$ to $\Lambda_j$. This allows us to rewrite the pressure equation as a linear system in $p = \{p_m\}$ and $v = \{v_{ij}\}$. This system takes the following form:

$$\begin{bmatrix} B & -C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} v \\ p \end{bmatrix} = \begin{bmatrix} 0 \\ q \end{bmatrix}. \quad \text{(106)}$$

47
Here, the $B$ and $C$ blocks are defined as follows:

$$b_{ijkl} = \int_{\Lambda} \psi_{ij} \cdot (\lambda K)^{-1} \psi_{kl} \, dx, \quad (107)$$

$$c_{m,kl} = \int_{\Lambda} \chi_m \nabla \cdot \psi_{kl} \, dx, \quad (108)$$

$$q_m = \int_{\Lambda} \chi_m q \, dx. \quad (109)$$

### 3.2.2. Discretizing the saturation equation

The saturation is evolved in time with a finite-volume scheme and an implicit time discretization. Consider a cell $\Lambda_i$ with edges $\gamma_{ij}$, then the finite-volume scheme takes the following form:

$$s_i^{n+1} = s_i^n + \frac{\Delta t}{\phi_i |\Lambda_i|} \left( Q_i(s_i^{n+1}) - \sum_{j=1}^{n_{equal}} F_{ij}(s_i^{n+1}) v_{ij} \right), \quad (110)$$

where $\Delta t$ denotes the time step and $|\Lambda_i|$ is the measure of grid cell $\Lambda_i$. $\phi_i$ is the porosity in $\Lambda_i$ and is a constant on the cell. $Q_i(s_i^{n+1}) = \int_{\Lambda_i} q_w(s_i^{n+1}) \, dx$ is the source contribution in $\Lambda_i$, and

$$F_{ij}(s_i^{n+1}) = \max \left\{ \text{sign}(v_{ij}) f_w(s_i^{n+1}), -\text{sign}(v_{ij}) f_w(s_j^{n+1}) \right\} \quad (111)$$

is the upstream-weighted fractional flow function at $\gamma_{ij}$, where $s_i^{n+1}$ is the saturation in $\Lambda_i$ at time step $t_{n+1}$.

A Newton-Raphson iterative method is employed to solve the implicit system of Eq. (110) as in [45]. The initial guess is chosen to be the saturation field from the previous time step.

Following [44], the system of Eqs. (97) and (99) is solved by a sequential splitting method (IMPES) [47]. The algorithm in brief is as follows: first, the saturation from the previous time step is used to compute the saturation-dependent variables, then the pressure equation of Eq. (97) is solved with the mixed FEM method. Next, the velocity is kept as a constant and used to evolve the saturation to the next time step with an upstream-weighted finite-volume method. The whole algorithm is summarized in Algorithm 4.

In the numerical example to follow (Section 3.2.4), we take $\mu_w = 0.3 \, cP$, $\mu_o = 3.0 \, cP$, $s_{wc} = s_{or} = 0.2$ and initial saturation $s_0 = s_{wc}$. Note, that in general, we set
Algorithm 4 IMPES for solving the two-phase flow problem

Initialize: Set $t = 0$, $s_{wc}$, $s_{or}$, $s_0 = s_{wc}$, $\mu_w$, $\mu_o$

Set $\Delta t_p$ and $\Delta t_s$

while $t \leq T$ do

Calculate $\lambda'_w$ and $\lambda'_o$ using Eq. (98)

Solving $p'$ and $v'$ from Eq. (106)

Set $t_1 = 0$

while $t_1 \leq \Delta t_p$ do

Solving $s_{wt}^{t_1}$ from Eq. (110) using $v'$

Set $t_1 = t_1 + \Delta t_s$

end while

Set $t = t + \Delta t_p$

end while

two time steps $\Delta t_p$ and $\Delta t_s$, one for the evolution of $p$ and $v$, and the other one for updating the saturation $s$, respectively. The pressure and velocity evolve slower than the saturation, therefore, in practice, we often set $\Delta t_p \gg \Delta t_s$.

3.2.3. Parametrization of uncertainty

In this work, we consider that the input uncertainty comes from both the permeability and porosity of the oil reservoir. In earlier studies [4, 5, 16], only the random permeability was considered. In general, the permeability and porosity are strongly correlated. This is apparent from the SPE-10 data set [48], as shown in Fig. [14]
To parameterize the uncertainty, we need to construct a stochastic input model based on available observations. Here, we use the SPE-10 data set \[48\], which is measured in the range of 1200×2200×170 (ft\(^3\)) and discretized in a regular Cartesian grid with 60×220×85. In this work, we decompose the observations into 340 reservoir domains with each one in a 60×54 grid. Each observation includes the log-permeability and log-porosity on the 60×54 grid. We assume that these 340 observations follow a second-order random field \(G(x, \omega)\). The Karhunen-Loève expansion (KLE) \[8\] is employed to approximate this random field with a finite-dimensional representation:

\[
G(x, \omega) = \mathbb{E}[G(x)] + \sum_{i=1}^{N} \sqrt{\lambda_i} \phi_i(x) \xi_i, \tag{112}
\]

where \(\{\xi_i\}_{i=1}^{N}\) are uncorrelated random variables here taken to follow the standard normal \(N(0, 1)\) distribution, \(\mathbb{E}[G(x)]\) is the mean of the 340 observations. The covariance function is obtained from these observations as well and then a eigenvalue problem is solved to compute the eigenfunctions \(\phi_i(x)\) and eigenvalues \(\lambda_i\).

We select the first 100 dimensions to describe the SPE-10 measurements, which preserves a total of 91% energy from Fig.\[15\]. Fig.\[16\] gives an example of simultaneously sampled permeability and porosity generated from the constructed stochastic input model.

---

**Figure 14**: Porous media flow - one measurement from the SPE-10 data set, layer one: (a) log-permeability; (b) porosity.
Figure 15: Porous media flow - The normalized energy plot for the SPE-10 measurements. Here, "normalized" means that each eigenvalue is divided by the sum of all eigenvalues.

Figure 16: Porous media flow - Sampled permeability and porosity field by the constructed stochastic input model: (a) log-permeability; (b) porosity.

3.2.4. Numerical results

The input stochastic dimension is set as 100. For a 2D problem, $d_s = 2$ and the input dimension of the model is $d = d_c + d_s + d_t = 103$. The responses under consideration are the velocity components, the pressure and the saturation. As discussed above, our deterministic flow solver is the source driven model, with an injection well at the bottom left and production well at the top right. Therefore, the source term $q$ in the pressure equation (Eq. (97)) is defined as a vector with only two non-zero components,
one on the bottom left cell and the other one on the top right cell. In principle, these two values can be freely set since they only affect the speed of saturation in the reservoir domain. In the SPE10 projection [48], the water injection rate was set as 5000 fluid barrel per day and the reservoir domain was fully saturated around 2000 days. Therefore, in this paper, we set $q = \pm 2$ for injection/production wells to mimic this process.

The time range considered is $[0, 2100]$ days. For each stochastic input $\xi$, the response is observed on $30 \times 27$ square spatial grid, and recorded every 30 days, i.e., $n_t = 70$. The hyper-parameters are selected as $\alpha = 1$, $\gamma_r = 10$, $\gamma_v = 100$, $u_0 = 0_d$, $R_0 = 10^{-3}I_d$, $W_0 = 10^{-3}I_d$ and $\nu_0 = d$. The truncation level $M$ is set to be 20,000. The model is trained with $n_\xi = 40, 80$ and 160 observations, where the stochastic input $\xi$ is randomly sampled from $\mathcal{N}(0, 1)^n$. The approximated posterior distribution of the Dirichlet process converges at thousands of clusters. This agrees with the findings in [25, 26] that with a large number of observations, the DP tends to converge at a state with a larger number of clusters (some of similar sizes). Decomposition over the spatial domain and temporal domain is also observed as expected. Due to the high-dimensionality of the input, it is hard to plot the decomposition of the space. One could only observe certain patterns in the data clustering. For the spatial region around the source (bottom-left), the algorithm tends to decompose the spatial domain rather than the temporal domain, and for other regions, it is the opposite. No specific patterns of the stochastic space were observed.

**Remark 7.** Several computational improvements can be considered in the way the local covariance matrix is calculated. For example, the separable framework in [5] can be easily integrated with this work to simplify the calculation of the local covariance matrix. This requires certain modifications in the calculation of the posterior distribution of $\theta_m$ and in the calculation of statistics.

After obtaining the approximations to the posterior distributions of interest from the variational inference algorithm, we draw 100 samples from them and construct the corresponding 100 surrogate models. For each sampled surrogate, the predictions are made on a $60 \times 54$ grid and every 20 days. The statistics of interest can be calculated
semi-analytically, using Eqs. (93) and (95). The mean and the standard deviation of the statistics of interest could be calculated by using the 100 sampled surrogates. The predictions are compared to the Monte Carlo estimates with $10^5$ observations. Fig. 17 compares the mean predictions of the mean of the saturation with different sizes of the training data set at $T = 1000$ days to the MC estimate. Two standard deviations (error bars) of the mean of the saturation for the case of $n_\xi = 160$ is provided in Fig. 17(d). Fig. 18 compares the mean predictions of the variance of the saturation at $T = 1000$ days to the MC estimate. The error bars for $n_\xi = 160$ are also provided. The same statistics are reported for the saturation at $T = 2000$ days in Figs. 19 and 20. Also, Figs. 21-24 show the mean predictions of the mean and standard deviation for the velocity components at $T = 2000$ days and compare them to the MC estimates. To allow reasonable visualization of the highly heterogeneous nature of the shown fields, the natural logarithm of the velocity components is plotted.
Figure 17: Porous media flow - comparison of mean predictions of the mean of the saturation at $T = 1000$ days provided by the model with (a) $n_{\xi} = 40$; (b) $n_{\xi} = 80$; (c) $n_{\xi} = 160$, to the MC result with (g) $N = 100,000$. Subfigure (d) shows the two standard deviations (error bars) of the mean of the saturation at $T = 1000$ days for $n_{\xi} = 160$ observations. Subfigures (e) and (f) show the difference between the predicted mean and MC estimate for $n_{\xi} = 80$ and 160, respectively.
Figure 18: Porous media flow - comparison of mean predictions of the std of the saturation at $T = 1000$ days provided by the model with (a) $n_{\xi} = 40$; (b) $n_{\xi} = 80$; (c) $n_{\xi} = 160$, to the MC result with (g) $N = 100,000$. Subfigure (d) shows the two standard deviations (error bars) of the std of the saturation at $T = 1000$ days for $n_{\xi} = 160$ observations. Subfigures (e) and (f) show the difference between the predicted std and MC estimate for $n_{\xi} = 80$ and 160, respectively.
Figure 19: Porous media flow - comparison of mean predictions of the mean of the saturation at $T = 2000$ days provided by the model with (a) $n_{\xi} = 40$; (b) $n_{\xi} = 80$; (c) $n_{\xi} = 160$, to the MC result with (g) $N = 100,000$. Subfigure (d) shows the two standard deviations (error bars) of the mean of the saturation at $T = 2000$ days for $n_{\xi} = 160$ observations. Subfigures (e) and (f) show the difference between the predicted mean and MC estimate for $n_{\xi} = 80$ and 160, respectively.
Figure 20: Porous media flow - comparison of mean predictions of the std of the saturation at $T = 2000$ days provided by the model with (a) $n_\xi = 40$; (b) $n_\xi = 80$; (c) $n_\xi = 160$, to the MC result with (g) $N = 100,000$. Subfigure (d) shows the two standard deviations (error bars) of the std of the saturation at $T = 2000$ days for $n_\xi = 160$ observations. Subfigures (e) and (f) show the difference between the predicted std and MC estimate for $n_\xi = 80$ and 160, respectively.
Figure 21: Porous media flow - comparison of mean of the mean of the natural log of the $x$-velocity component at $T = 2000$ days provided by the model with (a) $n_\xi = 40$, (b) $n_\xi = 80$, (c) $n_\xi = 160$, to the MC result with (e) $N = 100,000$. Subfigure (d) shows the two standard deviations (error bars) of the mean of the natural log of $x$-velocity component at $T = 2000$ days for $n_\xi = 160$ observations.
Figure 22: Porous media flow - comparison of mean predictions of the std of the natural log of the $x$-velocity component at $T = 2000$ days with (a) $n_{E} = 40$; (b) $n_{E} = 80$; (c) $n_{E} = 160$, to the MC result with (e) $N = 100,000$. Subfigure (d) shows the two standard deviations (error bars) of the std of the natural log of $x$-velocity component at $T = 2000$ days for $n_{E} = 160$ observations.
Figure 23: Porous media flow - comparison of mean of the mean of the natural log of the $y$-velocity at $T = 2000$ days provided by the model with (a) $n \xi = 40$; (b) $n \xi = 80$; (c) $n \xi = 160$, to the MC result with (e) $N = 100,000$. Subfigure (d) shows the two standard deviations (error bars) of the mean of the natural log of the $y$-velocity at $T = 2000$ days for $n \xi = 160$ observations.
Figure 24: Porous media flow - comparison of mean predictions of the std of the natural log of the y-velocity at $T = 2000$ days provided by the model with (a) $n_x = 40$; (b) $n_x = 80$; (c) $n_x = 160$, to the MC result with (e) $N = 100,000$. Subfigure (d) shows the two standard deviations (error bars) of the std of the natural log of the y-velocity component at $T = 2000$ days for $n_x = 160$ observations.

The calculation of the predicted probability densities is the same as for the KO problem. Figs. 25 and 26 show the predicted PDFs of the saturation at $T = 1000$ and $T = 2000$ days, respectively, at various spatial locations with different number of train-
ing data, and compare them to the MC estimates with $10^5$ observations. Subfigures (a) show the probability densities at location (10, 10) on the spatial grid. We can observe that the two distinct tails of the distribution are gradually captured by increasing the number of observations. This demonstrates that the proposed framework has a better performance in the prediction of PDFs compared to the results in [5]. Subfigure (b) plots the densities in the middle of the mesh. Note that there are no negative saturation values in the samples. A small peak around zero in Fig. 25(b) is simply given by the kernel density estimator which tends to provide a smooth representation. One can observe that with only $n_\xi = 160$ observations, the PDFs can be accurately captured.

Figure 25: Porous media flow - comparison of mean predictions of the PDFs of the saturation at various locations at $T = 1000$ days provided by the model with $N_\xi = 40, 80, 160$ to the MC results, (a), (b), (c) at location (10, 10); and (d), (e), (f) at location (30, 22).
Figure 26: Porous media flow - comparison of mean predictions of the PDFs of the saturation at various locations at \( T = 2000 \) days provided by the model with \( N_\xi = 40, 80, 160 \) to the MC results, (a) (b) (c) at location (10, 10); (d) (e) (f) at location (30, 22); (g) (h) (i) at location (5, 50).

The water-cut curve represents how much oil is produced at each time in the form of the fractional flow \( F(t) \) defined in Eq. (100). The prediction of the water-cut curve is calculated from the predicted velocity and saturation. The process of obtaining the predictions is the same as above. For each sampled surrogate, we calculate the mean and variance of the water-cut curve. This step is time consuming since the velocity and saturation at every \( t = 20 \) days needs to be predicted for the fractional flow calculation. Fig. 27(a) provides a comparison of the mean prediction of the mean water-cut curve with various number of observations to the MC estimates with \( 10^5 \) observations.
The comparison to MC of the standard deviation of the water-cut curve is given in Fig. 27(b). From these two figures, we can conclude that the mean of the water-cut curve can be easily captured, whereas the variance can be gradually captured as we increase the number of observations.

Figure 27: Porous media flow - comparison of the predictions of the water cut curve with different number of observations $N_{\xi} = 40, 80, 160$ to the MC results, (a) (b) (c) mean predictions of mean water cut and MC estimate; (d) (e) (f) mean predictions of std water cut and MC estimate.

4. Discussion and Conclusions

A Bayesian framework based on an infinite mixture of multi-output Gaussian processes was developed to address uncertainty propagation in various problems governed by differential equations. In the flow in random media problem, the input uncertainty was assumed to come from the subsurface permeability and porosity. The outputs of interest were the flow and pressure responses. The input variables considered in the proposed framework involved not only the stochastic variables, but also the spatial and time variables. The framework had the ability to capture the non-Gaussian or local
features due to the nature of the mixture model. The optimal number of mixture models could be automatically found by assigning a Dirichlet process prior. Each mixture component was one multi-output Gaussian process model which explained the local nonlinear relationship between the inputs and responses. The posterior distribution of interest was approximated by a variational inference algorithm. A probabilistic surrogate model was then constructed to give predictions for the statistics of interest.

Various examples were considered to study the accuracy and efficiency of the proposed framework. It was shown that this framework was capable of providing reliable predictions for the statistics with rather limited number of observations. In the provided examples, it was demonstrated that this framework could correctly provide the mean predictions of the first- and second-order statistics and reasonable error bars as well. The non-Gaussian feature of the PDFs was correctly captured. It was also shown that the evolution of the uncertainty propagation over time could be efficiently predicted. Various tasks remain to be further considered including (i) approaches that narrow the predicted error bars; (ii) exploring efficient decompositions of the covariance function for each GP in the mixture; (iii) more efficient treatment of the normalization constant in the calculation of responsibilities; and (iv) integrating this work with inverse problems solving (e.g. use limited output data to predict unobservable input information).

Acknowledgements

The work of NZ at the University of Warwick is supported by the Royal Society through a Wolfson Research Merit Award and from EPSRC (grant EP/L027682/1). In addition, NZ as Hans Fisher Senior Fellow acknowledges support of the Technische Universität München - Institute for Advanced Study, funded by the German Excellence Initiative and the European Union Seventh Framework Programme under grant agreement no 291763. The research at Cornell was supported by an OSD/AFOSR MURI09 award on uncertainty quantification, the US Department of Energy, Office of Science, Advanced Scientific Computing Research and the Computational Mathematics program of the National Science Foundation (NSF) (award DMS-1214282). This research used resources of the National Energy Research Scientific Computing Center,
which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. Additional computing resources were provided by the NSF through TeraGrid resources provided by NCSA under Grant No. TG-DMS090007.

Appendix A Derivation of the posterior of hyper-parameters

We sketch the proof of the posterior of hyper-parameters (Eq. (18)). We consider the $m$-th data subset but for mathematical convenience, the subscript $m$ is not shown. From Eq. (15), we write:

$$p(D|B, \Sigma, \theta) = \mathcal{N}_{n \times q}(Y|HB, A, \Sigma) = \frac{1}{(2\pi)^{\frac{nq}{2}}|\Sigma|^{\frac{nq}{2}}} \exp \left\{ -\frac{1}{2} \text{Tr}(\Sigma^{-1}(Y - HB)^T A^{-1}(Y - HB)) \right\}. \quad (113)$$

Using the priors for $B$, $\Sigma$ and $\theta$, $\pi(B, \Sigma, \theta) \propto \pi(B, \Sigma) \pi(\theta) \propto |\Sigma|^{-\frac{q+1}{2}}$, the posterior of $B$, $\Sigma$ and $\theta$ can be written as:

$$p(B, \Sigma, \theta|D) \propto p(D|B, \Sigma, \theta) \pi(B, \Sigma, \theta) \propto \pi(\theta)|\Sigma|^{-\frac{nq}{2}}|A|^{-\frac{n}{2}} \exp \left\{ -\frac{1}{2} \text{Tr}(\Sigma^{-1}(Y - HB)^T A^{-1}(Y - HB)) \right\}. \quad (114)$$

Now let us define $\Phi = (Y - HB)^T A^{-1}(Y - HB)$. Introducing $\hat{B} = (H^T A^{-1}H)^{-1}H^T A^{-1}Y$, we can simplify $\Phi$ as follows:

$$\Phi = (Y - HB)^T A^{-1}(Y - HB) = (B - \hat{B})^T (H^T A^{-1}H)(B - \hat{B}).$$

Substitution of this expression into Eq. (114) results in:

$$p(B, \Sigma, \theta|D) \propto \pi(\theta)|\Sigma|^{-\frac{nq}{2}}|A|^{-\frac{n}{2}} \exp \left\{ -\frac{1}{2} \text{Tr}(\Sigma^{-1}(Y - HB)^T A^{-1}(Y - HB)) \right\} \exp \left\{ -\frac{1}{2} \text{Tr}(\Sigma^{-1}(B - \hat{B})^T (H^T A^{-1}H)(B - \hat{B})) \right\}. \quad (115)$$

From this expression, we can immediately conclude Eq. (28) and also show that

$$B|D, \Sigma, \theta \sim \mathcal{N}_{n \times q}(\hat{B}, (H^T A^{-1}H)^{-1}, \Sigma) \quad (115)$$
where \( \hat{B} \) is given by Eq. (26).

Integrating \( B \) out of Eq. (114) gives:

\[
p(\Sigma, \theta|D) \propto \pi(\theta)\Sigma^{-\frac{q}{2}}|A|^{-\frac{q}{2}} \exp\left\{ -\frac{1}{2} \text{Tr}[\Sigma^{-1}(Y - \hat{H}\hat{B})^T A^{-1}(Y - \hat{H}\hat{B})]\right\} \\
\int \exp\left\{ -\frac{1}{2} \text{Tr}[(B - \hat{B})^T (H^TA^{-1}H)(B - \hat{B})]\right\} dB \\
\propto \pi(\theta)\Sigma^{-\frac{q}{2}}|A|^{-\frac{q}{2}} \exp\left\{ -\frac{1}{2} \text{Tr}[\Sigma^{-1}(Y - \hat{H}\hat{B})^T A^{-1}(Y - \hat{H}\hat{B})]\right\} \\
|\Sigma|^{-\frac{q}{2}}|H^TA^{-1}H|^{-\frac{q}{2}}|A|^{-\frac{q}{2}} \exp\left\{ -\frac{1}{2} \text{Tr}[\Sigma^{-1}(Y - \hat{H}\hat{B})^T A^{-1}(Y - \hat{H}\hat{B})]\right\}.
\]

(116)

From the above expression, we can verify that the posterior of \( \Sigma \) follows an inverse-Whishart distribution with \( q \) dimensions and \( n \) degrees of freedom:

\[
\Sigma|D, \theta \sim W_{q}^{-1}((n - p)\Sigma|W, n - p),
\]

(117)

where

\[
W = \frac{1}{n - p}(Y - \hat{H}\hat{B})^T A^{-1}(Y - \hat{H}\hat{B}).
\]

(118)

The posterior \( p(\theta|D, B, \Sigma) \) can be computed as \( p(\theta)p(D|B, \Sigma, \theta) \), where the likelihood \( p(D|B, \Sigma, \theta) \) is computed from Eq. (15) and the prior \( p(\theta) \) from Eq. (13). This is a computationally non-tractable posterior that in Section 2.4 is approximated by variational inference. The posteriors \( p(B|D, \Sigma, \theta) \), \( p(\Sigma|D, \theta) \) and \( p(\theta|D, B, \Sigma) \) are essential in the implementation of the variational inference algorithm of Section 2.4.
Appendix B  Variational Inference: Proof of Eq. (54)

We want to show that: \(\ln p(\mathcal{D}) = KL[q(\Psi)||p(\Psi|\mathcal{D})] + \mathcal{L}(q, \mathcal{D})\). Following simple algebra, one can show:

\[
\ln p(\mathcal{D}) = \ln p(\mathcal{D}, \Psi) - \ln p(\Psi|\mathcal{D}) = \int q(\Psi) \ln \frac{p(\mathcal{D}, \Psi)}{p(\Psi|\mathcal{D})} d\Psi
\]

\[
= \int q(\Psi) \ln \left( \frac{p(\mathcal{D}, \Psi)}{p(\Psi|\mathcal{D}) q(\Psi)} \right) d\Psi
\]

\[
= \int q(\Psi) \left( \ln \frac{q(\Psi)}{p(\Psi|\mathcal{D})} + \ln \frac{p(\mathcal{D}, \Psi)}{q(\Psi)} \right) d\Psi
\]

\[
= \int q(\Psi) \ln \frac{q(\Psi)}{p(\Psi|\mathcal{D})} d\Psi + \int q(\Psi) \ln \frac{p(\mathcal{D}, \Psi)}{q(\Psi)} d\Psi
\]

\[
= KL[q(\Psi)||p(\Psi|\mathcal{D})] + \mathcal{L}(q, \mathcal{D}).
\]

Appendix C  Variational Inference: Proof of Eq. (58)

We are interested to compute a distribution of the form \(q(\Psi) = \prod_k q_k(\omega_k)\) that maximizes the lower bound \(\mathcal{L}(q, \mathcal{D})\). Denoting for simplicity \(q_k(\omega_k) = q_k\), we can write:

\[
\mathcal{L}(q, \mathcal{D}) = \int \prod_k q_k \left[ \ln p(\Psi, \mathcal{D}) - \sum_k \ln q_k \right] d\Psi
\]

\[
= \int \prod_k q_k \ln p(\Psi, \mathcal{D}) \prod_k d\omega_k - \sum_k \int \prod_j q_j \ln q_j d\omega_j
\]

\[
= \int q_j \ln p(\Psi, \mathcal{D}) \prod_{k \neq j} (q_k d\omega_k) d\omega_j
\]

\[
- \int q_j \ln q_j d\omega_j - \sum_{k \neq j} \int q_k \ln q_k d\omega_k
\]

\[
= \int q_j \ln \frac{\exp E_{F_q(\omega_j) [\ln p(\Psi, \mathcal{D})]}}{q_j} d\omega_j - \sum_{k \neq j} \int q_k \ln q_k d\omega_k
\]

\[
= -KL(q_j || \exp E_{F_q(\omega_j) [\ln p(\Psi, \mathcal{D})]}) - \sum_{k \neq j} \int q_k \ln q_k d\omega_k.
\]

Clearly, the lower bound \(\mathcal{L}(q, \mathcal{D})\) is maximized when the Kullback-Leibler distance becomes zero, which is the case for \(\ln q_j(\omega_j) = E_{F_q(\omega_j) [\ln p(\Psi, \mathcal{D})]}\). The normalized
distribution is finally given as follows:

\[ q_j(\omega_j) = \frac{\exp \left( \mathbb{E}_{q(\omega_j)} [\ln p(\Psi, D)] \right)}{\int \exp \left( \mathbb{E}_{q(\omega_j)} [\ln p(\Psi, D)] \right) d\omega_j}. \]

Appendix D  Variational Inference: Proof of Eq. (83)

We start by using the Gaussian mixture approximation for \( q(\theta_m) \):

\[ \mathcal{H}[q] = \frac{1}{\sum_{l=1}^{L} \omega_l^2} \sum_{l=1}^{L} \omega_l^2 \int_{\theta_m} N(\theta_m; m_l, \sigma_l^2 I_{d+1}) \ln q(\theta_m) d\theta_m \]

\[ = -\frac{1}{\sum_{l=1}^{L} \omega_l^2} \sum_{l=1}^{L} \omega_l^2 \int_{\theta_m} N(\theta_m; m_l, \sigma_l^2 I_{d+1}) \ln \left( \frac{1}{\sum_{j=1}^{L} \omega_j^2} \sum_{j=1}^{L} \omega_j^2 N(\theta_m; m_j, \sigma_j^2 I_{d+1}) \right) d\theta_m. \]

Since \( -\ln(x) \) is concave in \( x \), using Jensen’s inequality, we can write \( -\ln(\mathbb{E}[x]) \leq \mathbb{E}[-\ln x] \). Using this result, one can show the following:

\[ \mathcal{H}[q] \geq -\frac{1}{\sum_{l=1}^{L} \omega_l^2} \sum_{l=1}^{L} \omega_l^2 \ln \left( \int_{\theta_m} N(\theta_m; m_l, \sigma_l^2 I_{d+1}) \left( \frac{1}{\sum_{j=1}^{L} \omega_j^2} \sum_{j=1}^{L} \omega_j^2 N(\theta_m; m_j, \sigma_j^2 I_{d+1}) \right) d\theta_m \right) \]

\[ = -\frac{1}{\left( \sum_{l=1}^{L} \omega_l^2 \right)^2} \sum_{l=1}^{L} \omega_l^2 \ln \left( \int_{\theta_m} \sum_{j=1}^{L} \omega_j^2 N(\theta_m; m_j, \sigma_j^2 I_{d+1}) \right) N(\theta_m; m_l, \sigma_l^2 I_{d+1}) d\theta_m \]

\[ = -\frac{1}{\left( \sum_{l=1}^{L} \omega_l^2 \right)^2} \sum_{l=1}^{L} \omega_l^2 \ln q_l. \]

The argument in the logarithm function above is denoted as \( q_l \) and it can be shown that it takes the following simplified form: \( q_l = \sum_{j=1}^{L} \omega_j^2 q_{lj} \), \( q_{lj} = N(m_l; m_j, (\sigma_l^2 + \sigma_j^2) I_{d+1}) \).

Indeed, we can prove this using the normalization of the multivariate Gaussian and simple algebra:
\[
q_l = \int \sum_{j=1}^{L} \omega_j^2 \mathcal{N}(\theta_m; m_j, \sigma_j^2 \mathbb{I}_{d+1}) \mathcal{N}(\theta_m; m_i, \sigma_i^2 \mathbb{I}_{d+1}) d\theta_m
\]
\[
= \sum_{j=1}^{L} \omega_j^2 \int_{\theta_m} \frac{1}{(2\pi \sigma_j^2)^{(d+1)/2}} \exp \left\{ -\frac{1}{2\sigma_j^2} (\theta_m - m_j)^T (\theta_m - m_j) \right\}
\]
\[
\frac{1}{(2\pi \sigma_j^2)^{(d+1)/2}} \exp \left\{ -\frac{1}{2\sigma_j^2} (\theta_m - m_i)^T (\theta_m - m_i) \right\} d\theta_m
\]
\[
= \sum_{j=1}^{L} \omega_j^2 \int_{\theta_m} \frac{1}{(2\pi \sigma_j^2)^{(d+1)/2}} \exp \left\{ -\frac{1}{2} \left[ \frac{\sigma_i^2 + \sigma_j^2}{\sigma_j^2 \sigma_i^2} \phi^{m_j} \right] \right\} d\theta_m
\]
\[
-2\theta^{(m_j)} \left( \frac{m_i}{\sigma_i^2} + \frac{m_j}{\sigma_j^2} \right) + \left( \frac{m_i^T m_i}{\sigma_i^2} + \frac{m_j^T m_j}{\sigma_j^2} \right) \right\} d\theta_m
\]
\[
= \sum_{j=1}^{L} \omega_j^2 \mathcal{N}(\theta_m; m_j, \sigma_j^2 \mathbb{I}_{d+1}).
\]

Appendix E  Variational Inference: Proof of Eq. (86) (Multivariate Delta Method for Moments)

Substituting Eq. (85) into Eq. (84) results in the following:

\[
\mathbb{E}_{\theta_m}[g(\theta_m)] \approx \frac{1}{\sum_{j=1}^{L} \omega_j^2} \sum_{j=1}^{L} \omega_j^2 \int_{\theta_m} \mathcal{N}(\theta_m; m_i, \sigma_i^2 \mathbb{I}_{d+1}) \tilde{g}(\theta_m) d\theta_m
\]
\[
= \frac{1}{\sum_{j=1}^{L} \omega_j^2} \sum_{j=1}^{L} \omega_j^2 \int_{\theta_m} \mathcal{N}(\theta_m; m_i, \sigma_i^2 \mathbb{I}_{d+1}) \left\{ g(m_i) + \nabla g(m_i)(\theta_m - m_i) + \frac{1}{2} (\theta_m - m_i)^T \mathcal{H}(\theta_m - m_i) \right\} d\theta_m
\]
\[
= \frac{1}{\sum_{j=1}^{L} \omega_j^2} \sum_{j=1}^{L} \omega_j^2 \left\{ g(m_i) + \int_{\theta_m} \mathcal{N}(\theta_m; m_i, \sigma_i^2 \mathbb{I}_{d+1}) \frac{1}{2} (\theta_m - m_i)^T \mathcal{H}(\theta_m - m_i) d\theta_m \right\}. \quad (119)
\]

Introducing \( \hat{\theta} = \theta_m - m_i \), we can rewrite the second term as:

\[
\mathbb{E}_{\theta_m} \left[ \frac{1}{2} (\theta_m - m_i)^T \mathcal{H}(\theta_m - m_i) \right] = \mathbb{E}_{\hat{\theta}} \left[ \frac{1}{2} \hat{\theta}^T \mathcal{H} \hat{\theta} \right]
\]
\[
= \frac{1}{2} \sum_i \mathbb{E}_{\hat{\theta}} \left[ \hat{\theta}_i \mathcal{H}_{ii} \hat{\theta}_i \right] + \frac{1}{2} \sum_{i \neq j} \mathbb{E}_{\hat{\theta}} \left[ \hat{\theta}_i \mathcal{H}_{ij} \hat{\theta}_j \right]
\]
\[
= \frac{1}{2} \sum_i \mathcal{H}_{ii} \mathbb{E}_{\hat{\theta}} \left[ \hat{\theta}_i \right] + \frac{1}{2} \sum_{i \neq j} \mathcal{H}_{ij} \mathbb{E}_{\hat{\theta}} \left[ \hat{\theta}_i \hat{\theta}_j \right]. \quad (120)
\]
Due to our representation in Eq. (74), we can derive that $E_{\theta} [\tilde{\theta} | \tilde{\theta}] = \sigma_i^2$, and for $i \neq j$, $E_{\theta} [\tilde{\theta} | \tilde{\theta}] = 0$. Substitution of these results in Eqs. (120) and (119) gives Eq. (86).

Appendix F  Variational Inference: Derivation of the Derivatives $\frac{\partial L_L[q]}{\partial \theta}$, $\frac{\partial L_L[q]}{\partial \sigma_k}$, and $\frac{\partial L_L[q]}{\partial \omega_k}$ of the Lower Bound

Starting from Eq. (83), we first calculate $\frac{\partial L_L[q]}{\partial m_i}$,

$$\frac{\partial L_L[q]}{\partial m_i} = \frac{1}{L} \sum_{k=1}^{L} \omega_k^2 \frac{\partial g(m_i)}{\partial m_i} - \frac{1}{(\sum_{k=1}^{L} \omega_k^2)^2} \sum_{k=1}^{L} \omega_k^2 \frac{\partial q_k}{\partial m_i}.$$  \hspace{1cm} (121)

Using the definition in Eq. (81) and Eqs. (113) and (13), we can write:

$$g(\theta_m) = \ln p(\theta_m, D_m)$$

$$= \ln \pi(\theta_m | \gamma) + \mathbb{E}_{B_m, \Sigma_m} [\ln p(D_m | \theta_m, B_m, \Sigma_m)]$$

$$\approx -\gamma \theta_m - \frac{1}{2} \ln |A_m| - \frac{1}{2} tr \left( \Sigma_m^{-1} \left( Y_m - H_m \hat{B}_m \right) A_m^{-1} (Y_m - H_m \hat{B}_m) \right) + \text{const}.$$  \hspace{1cm} (122)

Let $\bar{Y}_m = Y_m - H_m \hat{B}_m$. Taking the derivative of $g(\theta_m)$ w.r.t each component of $\theta \in \theta_m$, where $\theta_m = \{r_{m1}, \ldots, r_{md}, \epsilon_m\}$, results in:

$$\frac{\partial g(\theta_m)}{\partial \theta} \bigg|_{\theta_m} = -\gamma_0 - \frac{q}{2} tr \left( A_m^{-1} \frac{\partial A_m}{\partial \theta} \right) + \frac{1}{2} tr \left( \Sigma_m^{-1} \bar{Y}_m A_m^{-1} \frac{\partial A_m}{\partial \theta} A_m^{-1} \bar{Y}_m \right),$$

and

$$\frac{\partial g(\theta_m)}{\partial \theta} \bigg|_{\theta_m} = \left( \frac{\partial g(\theta_m)}{\partial r_{m1}}, \ldots, \frac{\partial g(\theta_m)}{\partial r_{md}}, \frac{\partial g(\theta_m)}{\partial \epsilon_m} \right) \bigg|_{\theta_m}.$$  \hspace{1cm} (123)

Now, let us look at the second term of Eq. (121). Recall that $q_k = \sum_{j=1}^{L} \omega_j^2 q_{kj}$ was defined in Eq. (83), where $q_{kj} = N(m_k, m_j, (\sigma_k^2 + \sigma_j^2) \mathbb{I})$,

$$\frac{\partial q_k}{\partial m_i} = \frac{\partial \sum_{j=1}^{L} \omega_j^2 q_{kj}}{\partial m_i} = \begin{cases} -\sum_{j=1}^{L} \omega_j^2 q_{kj} \frac{m_j - \mu_k}{\sigma_k^2}, & l = k \\ -\omega_j^2 q_{kl} \frac{m_j - \mu_k}{\sigma_j^2}, & l \neq k. \end{cases}$$  \hspace{1cm} (124)

This completes the calculation of all the terms needed in Eq. (121) to evaluate $\frac{\partial L_L[q]}{\partial \theta}$. Next, let us discuss how to calculate $\frac{\partial L_L[q]}{\partial \omega_k}$. Starting from Eq. (87), we can derive that:

$$\frac{\partial L_L[q]}{\partial \sigma_k} = \frac{1}{L} \sum_{i=1}^{L} \omega_i^2 \sigma_i tr (H_k) - \frac{1}{(\sum_{i=1}^{L} \omega_i^2)^2} \sum_{i=1}^{L} \omega_i^2 \frac{\partial q_i}{\partial \sigma_k}. $$  \hspace{1cm} (125)
For the first term, we can write:

\[ \text{tr}(\mathcal{H}_k) = \left( \frac{\partial^2 g(\theta_m)}{\partial \theta^2} + \sum_{j=1}^d \frac{\partial^2 g(\theta_m)}{\partial^2_{m,j}} \right)_{\theta_m}. \]

Each term of \( \theta \in \theta_m \) in the above equation is calculated as follows:

\[
\frac{\partial^2 g(\theta_m)}{\partial \theta} \bigg|_{\theta_m} = - \frac{d}{2} \text{tr} \left( A_{-1}^{-1} \frac{\partial^2 A_m}{\partial \theta^2} - A_m \frac{\partial A_m}{\partial \theta} A_m^{-1} \frac{\partial A_m}{\partial \theta} \right)
+ \frac{1}{2} \text{tr} \left( \Sigma_m^{-1} \left( \Sigma_m^{-1} \frac{\partial^2 A_m}{\partial \theta^2} A_m^{-1} \Sigma_m - 2 \Sigma_m^{-1} \frac{\partial A_m}{\partial \theta} A_m^{-1} \frac{\partial A_m}{\partial \theta} A_m^{-1} \Sigma_m \right) \right). 
\]

The second term in Eq. (122) can be obtained as:

\[
\frac{\partial q_i}{\partial \sigma_k} = \frac{\partial}{\partial \sigma_k} \sum_{j=1}^L \omega^2 q_i'_{kj}. 
\]

If \( l \neq k \), then

\[
\frac{\partial q_i}{\partial \sigma_k} = \frac{\partial (\omega^2 q_i'_{lk})}{\partial \sigma_k} = \omega^2 \frac{\partial}{\partial \sigma_k} \left[ \frac{(2\pi)^{-d/2} \sigma_k^{-d/2 - 1}}{\sigma_k} \right] \exp \left\{ -\frac{(m_k - m_l)^T (m_k - m_l)}{2(\sigma_k^2 + \sigma_l^2)} \right\} 
= \omega^2 \left[ (2\pi)^{-d/2} \sigma_k^{-d/2 - 1} \right] \exp \left\{ -\frac{(m_k - m_l)^T (m_k - m_l)}{2(\sigma_k^2 + \sigma_l^2)} \right\} 
+ (2\pi)^{-d/2} \sigma_k^{-d/2 - 1} \frac{\partial}{\partial \sigma_k} \left\{ -\frac{(m_k - m_l)^T (m_k - m_l)}{2(\sigma_k^2 + \sigma_l^2)} \right\} 
= \omega^2 \left[ (2\pi)^{-d/2} \sigma_k^{-d/2 - 1} \right] \left\{ -\frac{d + 1}{2(\sigma_k^2 + \sigma_l^2)} \right\} \exp \left\{ -\frac{(m_k - m_l)^T (m_k - m_l)}{2(\sigma_k^2 + \sigma_l^2)} \right\} 
+ (2\pi)^{-d/2} \sigma_k^{-d/2 - 1} \frac{\partial}{\partial \sigma_k} \left\{ -\frac{(m_k - m_l)^T (m_k - m_l)}{2(\sigma_k^2 + \sigma_l^2)} \right\} 
= \omega^2 \left[ -q_{ik}' \sigma_k \frac{d + 1}{2(\sigma_k^2 + \sigma_l^2)} \sigma_l \left\{ \frac{||m_k - m_l||_2}{(\sigma_k^2 + \sigma_l^2)} \right\} \right] 
= -\omega^2 q_{ik}' \sigma_k \left\{ \frac{d + 1}{2(\sigma_k^2 + \sigma_l^2)} - \frac{||m_k - m_l||_2}{(\sigma_k^2 + \sigma_l^2)} \right\}. 
\]

Similarly, we can derive the \( l = k \) case:

\[
\frac{\partial q_k}{\partial \sigma_k} = \frac{\partial}{\partial \sigma_k} \sum_{j=1}^L \omega^2 q_i'_{kj} 
= -\sum_{j=1}^L \omega^2 q_i'_{kj} \sigma_i \left\{ \frac{d + 1}{2(\sigma_k^2 + \sigma_l^2)} - \frac{||m_k - m_l||_2}{(\sigma_k^2 + \sigma_l^2)} \right\}. 
\]
Finally, to calculate $\frac{\partial \mathcal{L}[q]}{\partial \omega_k}$, we proceed as follows:

$$\frac{\partial \mathcal{L}[q]}{\partial \omega_k} = \frac{2\omega_k}{\sum_{l=1}^L \omega_l^2} \left[ g(m) + \frac{\sigma_l^2}{2} \text{tr}(H_l) \right]$$

$$- \frac{2\omega_k}{\left( \sum_{l=1}^L \omega_l^2 \right)^2} \sum_{l=1}^L \omega_l^2 \left[ g(m) + \frac{\sigma_l^2}{2} \text{tr}(H_l) \right]$$

$$+ \frac{4\omega_k}{\left( \sum_{l=1}^L \omega_l^2 \right)^3} \sum_{l=1}^L \omega_l^2 \ln q_l - \frac{1}{\left( \sum_{l=1}^L \omega_l^2 \right)^2} \sum_{l=1}^L \frac{\partial \omega_l^2 \ln q_l}{\partial \omega_k},$$

where

$$\frac{\partial \omega_l^2 \ln q_l}{\partial \omega_k} = \begin{cases} 2\omega_k \ln q_k, & l = k \\ \omega_l^2 \frac{\partial}{\partial q_l} q_l, & l \neq k. \end{cases}$$

References


77