Journal of Computational Physics 434 (2021) 110218

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A probabilistic generative model for semi-supervised training of coarse-grained surrogates and enforcing physical constraints through virtual observables

Maximilian Rixner, Phaedon-Stelios Koutsourelakis *

Professorship of Continuum Mechanics, Technical University of Munich, Germany

ARTICLE INFO

Article history:
Available online 22 February 2021

Keywords:
Probabilistic machine learning
Virtual observables
High-dimensional surrogates
Semi-supervised learning
Unlabeled data

ABSTRACT

The data-centric construction of inexpensive surrogates for fine-grained, physical models has been at the forefront of computational physics due to its significant utility in many-query tasks such as uncertainty quantification. Recent efforts have taken advantage of the enabling technologies from the field of machine learning (e.g., deep neural networks) in combination with simulation data. While such strategies have shown promise even in higher-dimensional problems, they generally require large amounts of training data even though the construction of surrogates is by definition a small data problem. Rather than employing data-based loss functions, it has been proposed to make use of the governing equations (in the simplest case, at collocation points) in order to imbue domain knowledge in the training of the otherwise black-box-like interpolators. The present paper provides a flexible, probabilistic framework that accounts for physical structure and information both in the training objectives as well as in the surrogate model itself. We advocate a *probabilistic* (Bayesian) model in which equalities that are available from the physics (e.g., residuals, conservation laws) can be introduced as *virtual* observables and can provide additional information through the likelihood. We further advocate a generative model i.e. one that attempts to learn the joint density of inputs and outputs that is capable of making use of *unlabeled* data (i.e., only inputs) in a semi-supervised fashion in order to reveal lower-dimensional embeddings of the high-dimensional input which are nevertheless predictive of the fine-grained model's output.

1. Introduction

The complexity and cost of many models in computational physics necessitates the development of less expensive surrogates (or coarse-grained/reduced-order models), which aim to emulate or approximate the mapping implicitly defined by the physical process between parametric inputs and the output at a significantly reduced cost. Such surrogates which retain sufficient predictive accuracy can be extremely valuable in *many-query* applications (e.g., inverse problems, uncertainty propagation, optimization) which would otherwise be inaccessible due to computational cost. The difficulty of constructing a suitable surrogate becomes particularly pronounced in the high-dimensional setting, i.e. when the number of input-output (random) variables is large as in most cases of practical interest. Data-based surrogates must also be capable of dealing with

* Corresponding author.
E-mail addresses: maximilian.rixner@tum.de (M. Rixner), p.s.koutsourelakis@tum.de (P.-S. Koutsourelakis).

https://doi.org/10.1016/j.jcp.2021.110218
0021-9991/

the scarcity of training data [1]. Unlike recent successes in statistical/machine learning, and supervised learning in particular, which in large part have been enabled by large datasets (and the computational means to leverage them), the acquisition of data, i.e. pairs of input-outputs, is the most expensive task and the reduction of their number, the primary objective of surrogate development.

Another critical challenge stems from the nature of the physical models themselves. Their primary utility arises from their ability to distill apparent complexity and high-dimensional descriptions into much fewer, essential variables and the relations between them, which can in turn be used to make accurate predictions under a variety of settings (e.g. different boundary/initial conditions, right-hand-sides). This robustness of physical models as well as their ability to operate under *extrapolative* conditions is not a property shared by black-box statistical surrogates, which in most cases are used in *interpolative* settings.

We put forward the proposition that to overcome these challenges, domain knowledge, i.e. information about the underlying physical/mathematical structure of the problem, must be injected into the surrogates constructed [2]. While this prior physical knowledge is generally plentiful and eloquently reflected in the governing equations, it is not necessarily obvious how to mine it, nor how to automatically combine it with the data-based learning objectives, especially in a probabilistic setting [3].

A probabilistic framework provides a superior setting for such problems as it is capable of quantifying predictive uncertainties which are unavoidable when any sort of model/dimensionality reduction is pursued and when the surrogate model is learned from finite (and hopefully, small) data [4].

The development of surrogates for the purposes of uncertainty quantification in the context of continuum thermodynamics where pertinent models are based on PDEs and ODEs has a long history. Some of the most well-studied methods have been based on (generalized) Polynomial Chaos expansions (gPC) [5,6] which have gained popularity due to the emergence of data-based, non-intrusive, sparse-grid stochastic collocation approaches [7–9]. These approaches typically struggle with high-dimensional stochastic inputs, as is the case, e.g. when random heterogeneous media [10] are considered.

Another strategy for the construction of inexpensive surrogates is offered by reduced-basis (RB) methods [11,12] where, based on a small set of "snapshots" ,i.e. input-output pairs, the solution space's dimensionality is reduced by projection onto the principal directions. Classical formulations rely on (Petrov-)Galerkin projections [13] for finding the associated coefficients, but recently several efforts have been directed towards unsupervised and supervised learning strategies [14–17]. Apart from issues of efficiency and stability, RB approaches in their standard form are generally treated in a non-Bayesian way and therefore only yield point estimates instead of full predictive posterior distributions. Furthermore, since scalar- or vector- or matrix-valued quantities need to be learned as a function of the parametric input in the offline phase, they are also challenged by the high-dimensions/small-data setting considered [18].

A more recent trend is to view surrogate modeling as a supervised learning problem and employ pertinent statistical learning tools, e.g. Gaussian Process (GP) regression [19–21], which can frequently provide closed-form predictive distributions. Although several advances have been made towards multi-fidelity data fusion [22–26] and incorporation of physical information [27–30] via Gaussian Processes, their performance and scaling with stochastic input dimension remains one of the main challenges. In the context of supervised learning, deep neural networks (DNNs) [31,32] have found their way into surrogate modeling of complex computer codes [33–37]. One of the most promising developments in the adaptation of such tools for physical modeling are physics-informed neural networks [38–41] which are trained by minimizing a loss function augmented by the residuals of the governing equations [42]. Physical knowledge in training DNNs has also been introduced in the form of residuals in [38,16,43–47] whereas in [48], a Boltzmann-type density containing physics-based functionals or residuals were employed as the target for the associated learning problem. Recent reviews of the use of various machine learning models, and in particular deep neural networks, for the solution of problems in computational physics, including the development of surrogates, can be found in [49,50]. Therein the difficulty of the task of incorporating physical domain-knowledge into machine learning objectives and tools [51,52] is detailed as well as the scarcity of probabilistic approaches in the context of such tasks.

In contrast to the majority of the efforts summarized above, our goal is not to provide a numerical discretization technique which aims to solve the PDE for a *single case*, but instead to learn the general input-output map defined by a parametric PDE. For this purpose, we consider as our reference model a discretized version of the PDE which is assumed to provide sufficiently accurate resolution (we refer to this as the Fine-Grained Model (FGM)). Furthermore, we wish to differentiate our work from applications of machine learning in problems where the underlying governing equations themselves are assumed unknown and one aims to identify them from data [53–55]. While a component of our model makes use of a (discretized) coarse-grained model, its form is in this work prescribed.

We propose overcoming the aforementioned challenges by introducing a novel, generative probabilistic model that is capable of exploiting labeled (i.e. input-output pairs) and unlabeled (i.e. only inputs) data in discovering lower-dimensional embeddings and identifying the right surrogate model-structure (section 2). More importantly, we propose augmenting the aforementioned data by injecting domain knowledge in a principled manner in the probabilistic models employed. In particular, such physical/mathematical knowledge is incorporated:

- in the learning objectives (section 2.2) through the novel notion of *virtual* observables [56]. We demonstrate how various types of information in the form of (non)linear equalities/constraints as well as minimizing functionals can be introduced in the likelihood terms.

- in an appropriately selected coarse-grained model (CGM, section 2.3) which through coarsened or reduced-physics versions of the full-order model provides an integral component of the proposed surrogate.

We complement the aforementioned elements with an integrated, supervised dimensionality reduction scheme which can distill lower-dimensional features of the high-dimensional input that are most predictive of the high-dimensional output and which is trained simultaneously with the other components by making use of (un)labeled data and virtual observables. We employ Stochastic Variational Inference techniques for training the proposed model (section 2.5), which yield a probabilistic surrogate that not only produces point estimates of the high-dimensional output but can quantify the predictive uncertainty associated with this task (section 2.6). We discuss the numerical complexity of the proposed algorithms in section 2.7 and assess the predictive performance of the proposed framework in section 3, where we demonstrate that unlabeled data and virtual observables can lead to significant improvements in its generalization accuracy and can reduce the number of labeled data (i.e., input-outputs pairs) to a few tens. Furthermore, we illustrate the model's ability to perform equally well under interpolative and extrapolative conditions, i.e., under boundary conditions seen or not seen during training. We finally demonstrate its benefits in an uncertainty propagation problem and discuss possible extensions in section 4.

2. Methodology

We illustrate the propose methodological framework in the context of steady-state, physical processes modeled by a partial differential equation and associated boundary conditions (i.e. a boundary value problem) of the general form

$$\begin{aligned} &\mathcal{L}(u(\boldsymbol{s},\boldsymbol{x});\boldsymbol{x}) = 0, && \text{for } \boldsymbol{s}\in\Omega \\ &\mathcal{B}(u(\boldsymbol{s},\boldsymbol{x});\boldsymbol{x}) = 0 && \text{for } \boldsymbol{s}\in\partial\Omega \end{aligned} \tag{1}$$

over the physical domain $\Omega\subset\mathbb{R}^d$. The differential $\mathcal{L}$ and boundary $\mathcal{B}$ operators depend on the random parameters $\boldsymbol{x}\in\mathbb{R}^{d_x}$ and so does the solution of the PDE $u(\boldsymbol{s},\boldsymbol{x})$. We denote by $\boldsymbol{y}\in\mathbb{R}^{d_y}$ discretized (with respect to $\boldsymbol{s}$) version of the latter and by $\boldsymbol{y}(\boldsymbol{x})$ the input-output map implied by any of the usual PDE-discretization schemes. The governing equations are complemented by boundary conditions which might depend on the parameters $\boldsymbol{x}$. We refer to this discretized model as *fine-grained model* (FGM). We are interested in FGMs that are computationally demanding, i.e. the number of forward model runs determines the cost of the analysis task of interest (e.g. forward or backward uncertainty propagation, optimization). Furthermore, the problems of interest are high-dimensional, i.e. $d_x, d_y >> 1$, as in most cases of practical interest. Our goal is to construct a surrogate with the *least* possible *labeled* data N_l, i.e. input-output pairs $\mathcal{D}_l = \{\boldsymbol{x}^{(i_l)}, \boldsymbol{y}^{(i_l)} = \boldsymbol{y}(\boldsymbol{x}^{(i_l)})\}_{i_l=1}^{N_l}$,[1] while still delivering sufficiently accurate predictions.

It is clear that in the *small data* setting learning a probabilistic surrogate $p(\boldsymbol{y}|\boldsymbol{x})$ is possible only if the problem is amenable to dimensionality reductions, i.e. there exists a lower-dimensional set of features[2] of $\boldsymbol{x}$ that are predictive of $\boldsymbol{y}$ and/or the latter itself lives in a lower-dimensional manifold. The simultaneous discovery of such lower-dimensional embeddings through a latent variable model was demonstrated in [57,58] where the sought density $p(\boldsymbol{y}|\boldsymbol{x})$ was approximated by

$$p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x}) = \int p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{z})\; p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\,\mathrm{d}\boldsymbol{z}\,, \tag{2}$$

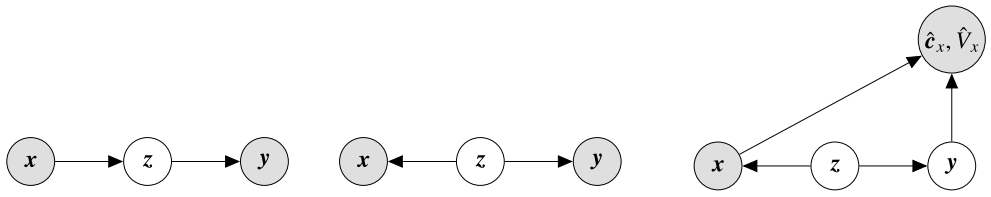
with $\boldsymbol{\theta}$ being the trainable parameters of the model. The variables $\boldsymbol{z}\in\mathbb{R}^Q$ represent the lower-dimensional (i.e. $Q << d_x, d_y$) information bottleneck between inputs and outputs. In the aforementioned works, these have been associated with a lower-fidelity physical model and have been identified in the presence of small data using sparse Bayesian learning from a large vocabulary of physically-motivated features of $\boldsymbol{x}$ (in contrast, in this work we will seek to identify predictive features of $\boldsymbol{x}$ purely based on data by making use of general blackbox function approximators, i.e. neural networks).

2.1. Generative model

The most direct approach in order to obtain a probabilistic surrogate would be to specify $p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})$ as is the case for wide array of methods. In the following we would like to suggest to the reader a different approach. The first novel contribution of this work is the use of a *generative* model, i.e. one that attempts to approximate the *joint* density $p(\boldsymbol{x},\boldsymbol{y})$ and which can subsequently be used by conditioning on $\boldsymbol{x}$ for predictive purposes. Such a model offers the capability to incorporate *unlabeled* data (i.e. only inputs) $\mathcal{D}_u = \{\boldsymbol{x}^{(i_u)}\}_{i_u=1}^{N_u}$ and therefore enables *semi-supervised* learning. This in turn allows the use of the information provided by the inexpensive (and potentially large) dataset $\mathcal{D}_u$ which can reduce the dependence on the expensive labeled data [59,60]. In particular, we propose a model that performs supervised dimensionality reduction of $\boldsymbol{x}$

[1] Each vector $\boldsymbol{y}^{(i_l)}$ is the discretized solution $u(\boldsymbol{s},\boldsymbol{x}^{(i_l)})$ of the governing PDE.

[2] i.e., there exist $d_\phi << \dim(\boldsymbol{x})$ functions $\{\phi_i(\boldsymbol{x})\}_{i=1}^{d_\phi}$ such that $p(\boldsymbol{y}|\boldsymbol{x}) \approx p\left(\boldsymbol{y}\middle|\{\phi_i(\boldsymbol{x})\}_{i=1}^{d_\phi}\right)$.



(a) Discriminative model (b) Generative model (c) Generative model *with virtual observables*

Fig. 1. Illustration of differences between probabilistic graphical models discussed (shaded nodes are observed). a) *Discriminative* model where the latent variables $\boldsymbol{z}$ encode lower-dimensional features of the input $\boldsymbol{x}$ which are predictive of the output $\boldsymbol{y}$, b) *Generative* model where $\boldsymbol{z}$ represent latent generators of both input and output, and c) *Generative* model which in comparison to (b) is augmented by *virtual observables* encoding domain knowledge.

and $\boldsymbol{y}$ [61] by postulating the existence of latent variables $\boldsymbol{z}$ that constitute $\boldsymbol{x}, \boldsymbol{y}$ *conditionally independent* (see Fig. 1b), i.e., for each labeled pair i_l in $\mathcal{D}_l$ the model assigns a likelihood

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i_l)}, \boldsymbol{y}^{(i_l)}) = \int p_{\boldsymbol{\theta}}(\boldsymbol{y}^{(i_l)}|\boldsymbol{z}^{(i_l)})\; p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i_l)}|\boldsymbol{z}^{(i_l)})\; p_{\boldsymbol{\theta}}(\boldsymbol{z}^{(i_l)})\; d\boldsymbol{z}^{(i_l)}\,. \tag{3}$$

We denote again with $\boldsymbol{\theta}$ any tunable model parameters, although these are in general different from the ones in Equation (2). The unobserved variables $\boldsymbol{z}$ play the role of latent generators of $\boldsymbol{x}$ and $\boldsymbol{y}$. We specify the form of the aforementioned densities, their parameterization as well as their training in the sequel. We note that the generative construction adopted provides also a likelihood for each unlabeled data point i_u in $\mathcal{D}_u$ as follows

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i_u)}) = \int p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i_u)}|\boldsymbol{z}^{(i_u)}) p_{\boldsymbol{\theta}}(\boldsymbol{z}^{(i_u)})\; d\boldsymbol{z}^{(i_u)}\,. \tag{4}$$

Furthermore, for predictive purposes, the posterior of $\boldsymbol{z}$ for a new $\boldsymbol{x}$, i.e. $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}) \propto p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) p_{\boldsymbol{\theta}}(\boldsymbol{z})$, can be used in order to compute

$$p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x}) = \int p_{\boldsymbol{\theta}}(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{x})\; d\boldsymbol{z} = \int p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{z}) p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\; d\boldsymbol{z}\,, \tag{5}$$

i.e., the predictive posterior on the corresponding output $\boldsymbol{y}$. Figs. 1a and 1b provide illustrations of the discriminative and generative probabilistic graphical models.

2.2. Virtual observables

The second novelty proposed in this paper pertains to the introduction of domain knowledge as represented in the governing equation (Equation (1)) into the learning objectives. We would like the training process not to rely exclusively on unlabeled $\mathcal{D}_u$ or labeled $\mathcal{D}_l$ data but also to incorporate physical knowledge. This can appear in several forms but since we are interested in their systematic incorporation we consider here various (in)equalities expressing different types of physical relations between the model-variables. The governing PDE of Equation (1) for example, is a potentially *infinite source* of information (if one considers that the equality holds at each of the infinite points of the problem domain Ω) in contrast to the limited times these governing equations can be solved due to computational expense. While the introduction of such equalities is rather straightforward in deterministic settings in the training loss and has been employed successfully in the context of physics-informed neural networks (PINNs [40]), in a probabilistic setting, it has only been achieved for linear ones and in order to approximate the solution of the PDE (not its dependence on input parameters) using Gaussian Processes [62]. In this work, we generalize the type of equalities that we consider by including nonlinear ones as well as demonstrate how other types of information, e.g. that the solution is a minimizer of a functional, can be incorporated. We discuss below how these can be integrated in the learning/inference process and we give specific examples of the forms these take in the numerical illustrations (section 3).

Consider first equality constraints, i.e.

$$\boldsymbol{c}(\boldsymbol{y}; \boldsymbol{x}) = \boldsymbol{0}\,, \tag{6}$$

where $\boldsymbol{c} : \mathbb{R}^{d_y} \times \mathbb{R}^{d_x} \to \mathbb{R}^{d_c}$. Such equalities can represent residuals of the governing PDE computed, e.g. at some collocation points or by employing weighted residuals with appropriate test/weight functions. They might also represent the enforcement of a physical constraint such as a conservation law (e.g. mass, momentum, energy). The only requirement on $\boldsymbol{c}$ imposed by our framework is that they are *differentiable* functions, a property that will prove crucial in the Stochastic Variational Inference component (section 2.5). In order to incorporate Equation (6), we introduce an auxiliary variable/vector $\hat{\boldsymbol{c}}_x$ which relates to $\boldsymbol{c}$ as follows

$$\hat{\boldsymbol{c}}_x = \boldsymbol{c}(\boldsymbol{y};\boldsymbol{x}) + \sigma_c \boldsymbol{\epsilon}_c, \qquad \boldsymbol{\epsilon}_c \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}) \tag{7}$$

We further assume that $\hat{\boldsymbol{c}}_x$ is *virtually observed* and $\hat{\boldsymbol{c}}_x = 0$. This induces a virtual likelihood $p(\hat{\boldsymbol{c}}_x|\boldsymbol{x}, \boldsymbol{y})$, i.e.

$$p(\hat{\boldsymbol{c}}_x = \boldsymbol{0} \mid \boldsymbol{x}, \boldsymbol{y}) \propto \frac{1}{\sigma_c^{d_c/2}}\, e^{-\frac{1}{2\sigma_c^2} \,||\, \boldsymbol{c}(\boldsymbol{y};\boldsymbol{x}) \,||_2^2}. \tag{8}$$

The parameter σ_c determines the intensity of the enforcement of the virtual observation and is analogous to the tolerance parameter with which constraints or residuals are enforced in deterministic solvers. In the limit that $\sigma_c \to 0$, the likelihood above degenerates to a Dirac-delta concentrated on the manifold implied by the constraint. In the context of the generative model proposed, one can exploit such unlabeled data, $\{\boldsymbol{x}^{(i_c)}, \hat{\boldsymbol{c}}_x^{(i_c)}\}$ consisting of pairs of inputs and *virtual observables* and the likelihood of each such data-pair i_c will be given by:

$$\begin{aligned} p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i_c)}, \hat{\boldsymbol{c}}_x^{(i_c)} = \boldsymbol{0}) &= \int p_{\boldsymbol{\theta}}(\hat{\boldsymbol{c}}_x^{(i_c)}, \boldsymbol{y}^{(i_c)}, \boldsymbol{z}^{(i_c)}, \boldsymbol{x}^{(i_c)})\, d\boldsymbol{y}^{(i_c)}\, d\boldsymbol{z}^{(i_c)} \\ &= \int p(\hat{\boldsymbol{c}}_x^{(i_c)} = \boldsymbol{0} \,|\boldsymbol{y}^{(i_c)}, \boldsymbol{x}^{(i_c)}) p_{\boldsymbol{\theta}}(\boldsymbol{y}^{(i_c)}, \boldsymbol{z}^{(i_c)}, \boldsymbol{x}^{(i_c)})\, d\boldsymbol{y}^{(i_c)}\, d\boldsymbol{z}^{(i_c)} \\ &= \int p(\hat{\boldsymbol{c}}_x^{(i_c)} = \boldsymbol{0} \,|\boldsymbol{y}^{(i_c)}, \boldsymbol{x}^{(i_c)}) p_{\boldsymbol{\theta}}(\boldsymbol{y}^{(i_c)}|\boldsymbol{z}^{(i_c)}) p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i_c)}|\boldsymbol{z}^{(i_c)})\; p_{\boldsymbol{\theta}}(\boldsymbol{z}^{(i_c)})\, d\boldsymbol{y}^{(i_c)}\, d\boldsymbol{z}^{(i_c)} \end{aligned} \tag{9}$$

We emphasize that in this case, the solution vector $\boldsymbol{y}^{(i_c)}$ (which satisfies the constraint $\boldsymbol{c}(\boldsymbol{y}^{(i_c)}; \boldsymbol{x}^{(i_c)})$) is latent and must be inferred. We also note that $\hat{\boldsymbol{c}}_x^{(i_c)} = 0$ in Equation (9) does *not* imply that we have conditioned on this observation, but that $\hat{\boldsymbol{c}}_x^{(i_c)}$ is always assumed to be (pseudo-) observed as equal to zero, and just like $\boldsymbol{x}^{(i_c)}$, is treated as observed data. The corresponding graphical model is illustrated in Fig. 1c where the virtual observables are depicted as observed nodes [63] with $\boldsymbol{y}$, the solution of the PDE, becoming a latent variable and therefore unknown quantity in this case.

Another type of physical information that can be accommodated with the concept of virtual observables pertains to the variational nature of the associated problem. It is well-known that the solutions of most PDEs in computational physics can be expressed as minimizers of appropriate functionals. Such functionals have served as the foundation of several numerical schemes and appear in various forms, even for irreversible, nonlinear processes [64,65]. Various versions of these functionals were incorporated in the machine-learning loss functions of deterministic, deep models [66] as well as in the likelihood functions of probabilistic models [48].

Suppose that the discretized solution vector $\boldsymbol{y}(\boldsymbol{x})$ is obtained as the minimizer of

$$\boldsymbol{y}(\boldsymbol{x}) = \arg\min_{\boldsymbol{y}} V(\boldsymbol{y};\boldsymbol{x}), \tag{10}$$

where $V : \mathbb{R}^{d_y} \times \mathbb{R}^{d_x} \to \mathbb{R}$ represents a generalized free energy or potential. Let $V_{min}(\boldsymbol{x}) = \min_{\boldsymbol{y}} V(\boldsymbol{y};\boldsymbol{x})$ be the unknown minimum value of V (attained by the solution) for each $\boldsymbol{x}$. We define a new, auxiliary variable $\hat{V}_x$ as

$$\hat{V}_x = V(\boldsymbol{y};\boldsymbol{x}) - V_{min}(\boldsymbol{x}) - \epsilon_V, \qquad \epsilon_V \sim \text{Expon}(\beta^{-1}). \tag{11}$$

The random variable ϵ_V is by construction always non-negative and follows an exponential distribution with parameter β.[3] We further assume that $\hat{V}_x = 0$ has been *virtually* observed which implies a *virtual likelihood*

$$p(\hat{V}_x = 0 \mid \boldsymbol{y}, \boldsymbol{x}) = \beta^{-1}\, e^{-\beta^{-1}(V(\boldsymbol{y};\boldsymbol{x}) - V_{min}(\boldsymbol{x}))}. \tag{12}$$

As it will be become clear in the sequel, the unknown $V_{min}(\boldsymbol{x})$ does not enter the training of the model. One can deduce from Equation (12) that the smaller $V(\boldsymbol{y};\boldsymbol{x})$ is, the higher the corresponding likelihood becomes and the latter is maximized for the $\boldsymbol{y}$ that corresponds to the solution (Equation (10)). Furthermore, the parameter β dictates the decay of the likelihood for $V(\boldsymbol{y};\boldsymbol{x}) > V_{min}(\boldsymbol{x})$ and in the limit $\beta^{-1} \to 0$, the likelihood degenerates to a Dirac-delta concentrated at the minimum (i.e. the true solution).

As in the previous case of the equality constraints, the introduction of these new observables enables the incorporation of the information contained in the discretized functional V in the training of the proposed generative model. In particular, given unlabeled data $\{\boldsymbol{x}^{(i_V)}, \hat{V}_x^{(i_V)}\}$ consisting of pairs of inputs and *virtual observables* $\hat{V}_x$, the likelihood implied by the model for each data-pair i_V will be:

$$\begin{aligned} p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i_V)}, \hat{V}_x^{(i_V)} = 0) &= \int p_{\boldsymbol{\theta}}(\hat{V}_x^{(i_V)} = 0,\; \boldsymbol{y}^{(i_V)}, \boldsymbol{z}^{(i_V)}, \boldsymbol{x}^{(i_V)})\, d\boldsymbol{y}^{(i_V)}\, d\boldsymbol{z}^{(i_V)} \\ &= \int p(\hat{V}_x^{(i_V)} = 0 \mid \boldsymbol{y}^{(i_V)}, \boldsymbol{x}^{(i_V)})\; p_{\boldsymbol{\theta}}(\boldsymbol{y}^{(i_V)}, \boldsymbol{z}^{(i_V)}, \boldsymbol{x}^{(i_V)})\, d\boldsymbol{y}^{(i_V)}\, d\boldsymbol{z}^{(i_V)} \\ &= \int p(\hat{V}_x^{(i_V)} = 0 \mid \boldsymbol{y}^{(i_V)}, \boldsymbol{x}^{(i_V)})\; p_{\boldsymbol{\theta}}(\boldsymbol{y}^{(i_V)}|\boldsymbol{z}^{(i_V)}) p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i_V)}|\boldsymbol{z}^{(i_V)})\; p_{\boldsymbol{\theta}}(\boldsymbol{z}^{(i_V)})\, d\boldsymbol{y}^{(i_V)}\, d\boldsymbol{z}^{(i_V)} \end{aligned} \tag{13}$$

As in Equation (9), the solution vector $\boldsymbol{y}^{(i_V)}$ (which minimizes $V(\boldsymbol{y};\boldsymbol{x}^{(i_V)})$ is latent and must be inferred.

To make our presentation independent of specific choices, in the remainder we denote a dataset of virtual observables by $\mathcal{D}_{\mathcal{O}} = \{\boldsymbol{x}^{(i_{\mathcal{O}})}, \hat{\boldsymbol{o}}^{(i_{\mathcal{O}})}\}_{i=1}^{N_{\mathcal{O}}}$, where $\boldsymbol{x}^{(i_{\mathcal{O}})}$ represents an *input query point* and the corresponding $\hat{\boldsymbol{o}}^{(i_{\mathcal{O}})} \in \mathbb{R}^M$ comprises the

[3] ϵ_V can be thought as the probabilistic analogue of a slack variable for the enforcement of inequality constraints in optimization.

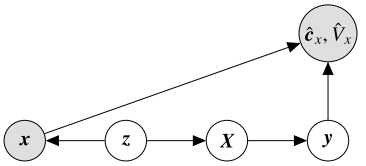


Fig. 2. Node $\boldsymbol{X}$ corresponds to the inputs of a deterministic coarse-grained model (CGM), implying that $\boldsymbol{z}$ is encouraged not only to learn a representation of the inputs $\boldsymbol{x}$, but also features that through the CGM can be predictive of the FGM output $\boldsymbol{y}$ (compare with Fig. 1c - shaded nodes are observed).

corresponding virtually observed values. Without loss of generality, we assume that we enforce the same number of M constraints at every point (this assumption can easily be relaxed). Parameters that govern how rigidly the constrains are enforced, such as σ_c^{-1} or β, are denoted summarily by $\boldsymbol{\tau}$; in the more general case, different constraints can be enforced to varying degrees, i.e. $\boldsymbol{\tau}$ can comprise several precision-type parameters and may be a vector instead of a scalar. We stress that the parameters $\boldsymbol{\tau}$ are conceptually different from the parameters $\boldsymbol{\theta}$ of the generative model, since they do not pertain to the generative process of $(\boldsymbol{x}, \boldsymbol{y})$, but rather govern the enforcement of physical constraints. In order to simplify the discussion and our notation, in the following we will assume that $\boldsymbol{\tau}$ is a-priori specified and therefore we will omit to explicitly condition on $\boldsymbol{\tau}$ (we discuss in Appendix C how $\boldsymbol{\tau}$ could be inferred if not known a-priori by introducing a variational approximation $q(\boldsymbol{\tau})$). We use the term *input query point* for each $\boldsymbol{x}^{(i_{\mathcal{O}})}$ appearing in $\mathcal{D}_{\mathcal{O}}$ to emphasize that in the general case the corresponding solution of the PDE $\boldsymbol{y}(\boldsymbol{x})$ is *not* observed/known, and we only *query* certain information from the underlying physics. The introduction of virtual observables implies that the plausibility of each model contained within the hypothesis space of the generative model $p_{\boldsymbol{\theta}}(\boldsymbol{y}, \boldsymbol{x})$ is scored not only according to its performance on unlabeled and labeled data, but also with respect to the associated physical constraints.

2.3. Physics-inspired structure for surrogate

The third contribution of the paper in the direction of imbuing physical knowledge into the machine learning framework pertains to the meaning of the latent variables $\boldsymbol{z}$ and the density $p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{z})$. While one can make use of a purely statistical model by employing, e.g., a Gaussian Process or a (deep) neural network, we advocate here building the surrogate around a *coarse-grained model* (CGM). The latter can be based on simply coarsening the discretization of the governing equations ([57]) or by employing simplified physics ([58]). It serves as a stencil that automatically retains the primary physical characteristics of the FGM and can therefore lead to a reduction of the amount of data needed for training.

Let $\boldsymbol{X}$ and $\boldsymbol{Y}$ denote the input and output vector of the aforementioned CGM. The physical meaning of these variables does not need to be the same as for $\boldsymbol{x}$ or $\boldsymbol{y}$ but they are, by construction, lower-dimensional and the solution of the CGM, i.e. the cost of each evaluation of $\boldsymbol{Y}(\boldsymbol{X})$[4] is negligible as compared to $\boldsymbol{y}(\boldsymbol{x})$. We propose:

- linking the latent features $\boldsymbol{z}$ with $\boldsymbol{X}$ through a density $p_{\boldsymbol{\theta}}(\boldsymbol{X}|\boldsymbol{z})$ with tunable parameters $\boldsymbol{\theta}$
- linking the sought FGM output $\boldsymbol{y}$ with the output of the CGM $\boldsymbol{Y}(\boldsymbol{X})$ rather than with $\boldsymbol{z}$ directly. Hence instead of $p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{z})$ we propose employing a density

$$p_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{Y}(\boldsymbol{X})) \tag{14}$$

These two elements combined allow us to express $p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{z})$ in Equation (5) as

$$p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{z}) = \int p_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{Y}(\boldsymbol{X}))\, p_{\boldsymbol{\theta}}(\boldsymbol{X}|\boldsymbol{z})\, d\boldsymbol{X}$$

and the (analytically intractable) predictive conditional density $p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})$ becomes

$$p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x}) = \int p_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{Y}(\boldsymbol{X}))\, p_{\boldsymbol{\theta}}(\boldsymbol{X}|\boldsymbol{z})\, p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\, d\boldsymbol{X}\, d\boldsymbol{z}\,. \tag{15}$$

By mapping to the CGM input $\boldsymbol{X}$, the latent variables $\boldsymbol{z}$, learn to reconstruct the FGM's solution $\boldsymbol{y}$ from the output $\boldsymbol{Y}$ of the CGM by means of $p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{Y}(\boldsymbol{X}))$ (Fig. 2).

We specify $\boldsymbol{X}, \boldsymbol{Y}$, the CGM itself as well as the densities involved in subsequent sections and in particular in the context of the numerical illustrations (section 3). The introduction of the CGM and the associated latent variables $\boldsymbol{X}$ (and $\boldsymbol{Y}$ for a stochastic CGM) does not alter the generative nature of the model. We note though that the CGM can be omitted or simply complemented by a phenomenological statistical emulator, in which case the graphical model structure in Fig. 2 would be altered.

[4] We assume a deterministic CGM for simplicity although this can be relaxed.

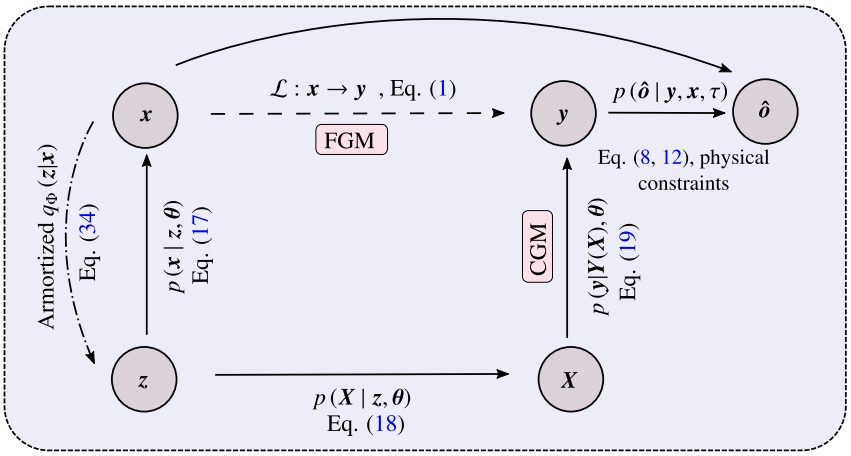


Fig. 3. A schematic overview of the building blocks of the generative model. All solid black arrows correspond to the conditional densities Eq. (17)–(19), i.e. encode conditional dependence assumptions, and therefore define the joint distribution $p_{\boldsymbol{\theta}}\left(\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{X}, \boldsymbol{y}, \hat{\boldsymbol{o}}\right) = p(\boldsymbol{z})\, p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\theta})\, p(\boldsymbol{X}|\boldsymbol{z}, \boldsymbol{\theta})\, p(\boldsymbol{y}|\boldsymbol{Y}(\boldsymbol{X}), \boldsymbol{\theta})\, p\left(\hat{\boldsymbol{o}}|\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\tau}\right)$. The dashed lines correspond to the amortized encoder (Eq. (34)) as an auxiliary tool for inference, as well as the mapping $\boldsymbol{y}(\boldsymbol{x})$ implied by the fine-scale resolution of the differential operator $\mathcal{L}$. The latent space encoding $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ (Eq. (16)) is assumed to have given rise to all other observed quantities via a series of conditional densities involving complex, parametric nonlinear transformations defined by $\boldsymbol{\theta}$ and the CGM $\boldsymbol{Y}(\boldsymbol{X})$. Since the latent dimension $Q = \dim(\boldsymbol{z})$ is considerably smaller than $d_x = \dim(\boldsymbol{x}), d_y = \dim(\boldsymbol{y})$, this implies that the model (via an information-bottleneck) has to identify a lower-dimensional embedding of the data $(\boldsymbol{x}, \boldsymbol{y})$ defined by $p(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta})$, which in turn is used to derive effective properties $\boldsymbol{X}$ via $p(\boldsymbol{X}|\boldsymbol{z}, \boldsymbol{\theta})$ (see Eq. (18)) entering the coarse-grained model $\boldsymbol{Y}(\boldsymbol{X})$; subsequently the predictions of the CGM are used to reconstruct the fine-scale solution via $p(\boldsymbol{y}|\boldsymbol{Y}(\boldsymbol{X}), \boldsymbol{\theta})$, see Eq. (19). If any of the nodes in this graph are observed, we can probabilistically reason about the parameters $\boldsymbol{\theta}$ that have given rise to these observations (using variational inference, see section 2.5). It is possible to leverage any kind of data (unlabeled, labeled, domain knowledge) to reason about $\boldsymbol{\theta}$ (by optimizing the combined ELBO Eq. (28)), and thereby identifying a suitable coarse-grained physics model in conjunction with some latent encoding out of an a-priori defined parametric family of candidates.

2.4. Specification of generative model

In the following we suggest a specific architecture for the probabilistic model which satisfies all of the previously discussed key aspects; i.e., a generative model that implicitly defines (and learns) a *joint* distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{y})$ via unobserved, latent variables $\boldsymbol{z}$ (see Eq. (3)), and where predictions for $\boldsymbol{y}$ are obtained by identifying a coarse-grained physical process based on the latent space encoding via the densities $p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{Y}(\boldsymbol{X}))$ and $p_{\boldsymbol{\theta}}(\boldsymbol{X}|\boldsymbol{z})$ (see Eq. (14), (15)). Assuming real-valued $\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{X}, \boldsymbol{y}$ we propose the following probabilistic generative model (for a schematic overview see also Fig. 3)

$$\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}) \tag{16}$$

$$\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{z}; \boldsymbol{\theta}_x) + \boldsymbol{S}_x^{1/2}(\boldsymbol{z}; \boldsymbol{\theta}_x)\, \boldsymbol{\varepsilon}_x \qquad \boldsymbol{\varepsilon}_x \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}) \tag{17}$$

$$\boldsymbol{X} = \boldsymbol{g}\left(\boldsymbol{z}; \boldsymbol{\theta}_g\right) + \boldsymbol{S}_X^{1/2} \boldsymbol{\varepsilon}_X \qquad \boldsymbol{\varepsilon}_X \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}) \tag{18}$$

$$\boldsymbol{y} = \boldsymbol{h}\left(\boldsymbol{Y}(\boldsymbol{X}); \boldsymbol{\theta}_y\right) + \boldsymbol{S}_y^{1/2} \boldsymbol{\varepsilon}_y \qquad \boldsymbol{\varepsilon}_y \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}) \tag{19}$$

where $\boldsymbol{f}(\cdot)$ and $\boldsymbol{g}(\cdot)$ are nonlinear functions (e.g. neural networks) parameterized by $\boldsymbol{\theta}_x$ and $\boldsymbol{\theta}_g$ respectively. We have assumed here a Gaussian noise model, implicitly parameterized by a set of symmetric positive definitive matrices $\boldsymbol{S}_x$, $\boldsymbol{S}_X$ and $\boldsymbol{S}_y$.[5] We defer any further discussion of the specifics until section 3 where the meaning of the different variables is presented. Since we operate under the assumption of *small labeled* data, the complexity of $\boldsymbol{g}(\boldsymbol{z}; \boldsymbol{\theta}_g \cdot)$ is chosen relatively low compared to $\boldsymbol{f}(\boldsymbol{z}; \boldsymbol{\theta}_x)$, in order to allow learning a mapping from latent space to effective properties $\boldsymbol{X}$ with comparably few examples. The role of $\boldsymbol{h}(\boldsymbol{Y}(\boldsymbol{X}); \boldsymbol{\theta}_y)$ is to define the map from the CGM's output $\boldsymbol{Y}(\boldsymbol{X})$ to the (mean of the) output $\boldsymbol{y}$ of the FGM. All the conditional densities in (17) - (19) are multivariate Gaussians which have constant covariances with the exception of Equation (17) where the covariance $\boldsymbol{S}_x$ depends on the $\boldsymbol{z}$ variables as dictated by the associated parameters $\boldsymbol{\theta}_x$.

We denote by $\boldsymbol{\theta} = \left\{\boldsymbol{\theta}_x, \boldsymbol{\theta}_g, \boldsymbol{\theta}_y, \boldsymbol{S}_X, \boldsymbol{S}_{\boldsymbol{y}}\right\}$ the parameters of the generative model, which we wish to learn from a dataset $\mathcal{D} = \left\{\mathcal{D}_u, \mathcal{D}_l, \mathcal{D}_{\mathcal{O}}\right\}$ which, in the most general case, consists of N_u unlabeled examples $\mathcal{D}_u = \left\{\boldsymbol{x}^{(i_u)}\right\}_{i_u=1}^{N_u}$, N_l labeled input-output examples $\mathcal{D}_l = \{(\boldsymbol{x}^{(i_l)}, \boldsymbol{y}^{(i_l)})\}_{i_l=1}^{N_l}$, and a collection $\mathcal{D}_{\mathcal{O}} = \{\boldsymbol{x}^{(i_{\mathcal{O}})}, \hat{\boldsymbol{o}}^{(i_{\mathcal{O}})}\}_{i_{\mathcal{O}}=1}^{N_{\mathcal{O}}}$ of $N_{\mathcal{O}}$ query input points and virtual observables. We may then write the marginal likelihood as

[5] We adopted a heteroscedastic noise model for $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ due to $\boldsymbol{S}_x(\boldsymbol{z}; \boldsymbol{\theta}_x)$ depending on the latent variables, while $\boldsymbol{S}_X$ and $\boldsymbol{S}_y$ are assumed constant. This difference in the noise models was necessitated by the fact that the identification of a heteroscedastic noise model requires (much) larger amounts of data, and we wish to operate (in the 'supervised' branch of the model) in the *small data* regime.

$$
\begin{aligned}
p\left(\mathcal{D}|\boldsymbol{\theta}\right) &= p(\mathcal{D}_u|\boldsymbol{\theta})\; p(\mathcal{D}_l|\boldsymbol{\theta})\; p(\mathcal{D}_{\mathcal{O}}|\boldsymbol{\theta}) \\
&= \prod_{i_u=1}^{N_u} p(\boldsymbol{x}^{(i_u)}|\boldsymbol{\theta})\; \prod_{i_l=1}^{N_l} p(\boldsymbol{x}^{(i_l)},\boldsymbol{y}^{(i_l)}|\boldsymbol{\theta})\; \prod_{i_{\mathcal{O}}=1}^{N_{\mathcal{O}}} p(\boldsymbol{x}^{(i_{\mathcal{O}})},\hat{\boldsymbol{o}}^{(i_{\mathcal{O}})}|\boldsymbol{\theta})\;,
\end{aligned}
\tag{20}
$$

where each of the likelihood terms in the products are given by Equations (4), (3) and (9) (or (13)) respectively. In view of the densities in Equations (16) - (19) these become

$$
p(\boldsymbol{x}^{(i_u)}|\boldsymbol{\theta}) = \int \mathcal{N}\left(\boldsymbol{x}^{(i_u)}|\; \boldsymbol{f}\left(\boldsymbol{z}^{(i_u)};\boldsymbol{\theta}_x\right),\; \boldsymbol{S}_x\left(\boldsymbol{z}^{(i_u)};\boldsymbol{\theta}_x\right)\right)\; \mathcal{N}\left(\boldsymbol{z}^{(i_u)}|\; \boldsymbol{0},\boldsymbol{I}\right)\; d\boldsymbol{z}^{(i_u)}\;,
\tag{21}
$$

$$
\begin{aligned}
p(\boldsymbol{x}^{(i_l)},\boldsymbol{y}^{(i_l)}|\boldsymbol{\theta}) &= \int \mathcal{N}\left(\boldsymbol{y}^{(i_l)}|\; \boldsymbol{h}\left(\boldsymbol{Y}\left(\boldsymbol{X}^{(i_l)}\right);\boldsymbol{\theta}_y\right),\boldsymbol{S}_y\right)\; \mathcal{N}\left(\boldsymbol{X}^{(i_l)}|\; \boldsymbol{g}\left(\boldsymbol{z}^{(i_l)};\boldsymbol{\theta}_g\right),\boldsymbol{S}_X\right) \\
&\qquad \mathcal{N}\left(\boldsymbol{x}^{(i_l)}|\; \boldsymbol{f}\left(\boldsymbol{z}^{(i_l)};\boldsymbol{\theta}_x\right),\; \boldsymbol{S}_x\left(\boldsymbol{z}^{(i_l)};\boldsymbol{\theta}_x\right)\right)\; \mathcal{N}\left(\boldsymbol{z}^{(i_l)}|\; \boldsymbol{0},\boldsymbol{I}\right)\; d\boldsymbol{X}^{(i_l)}\; d\boldsymbol{z}^{(i_l)}\;,
\end{aligned}
\tag{22}
$$

and

$$
\begin{aligned}
p(\boldsymbol{x}^{(i_{\mathcal{O}})},\hat{\boldsymbol{o}}^{(i_{\mathcal{O}})}|\boldsymbol{\theta}) &= \int p(\hat{\boldsymbol{o}}^{(i_{\mathcal{O}})}|\boldsymbol{y}^{(i_{\mathcal{O}})},\boldsymbol{x}^{(i_{\mathcal{O}})};\boldsymbol{\tau})\; \mathcal{N}\left(\boldsymbol{y}^{(i_{\mathcal{O}})}|\; \boldsymbol{h}\left(\boldsymbol{Y}\left(\boldsymbol{X}^{(i_{\mathcal{O}})}\right);\boldsymbol{\theta}_y\right),\boldsymbol{S}_y\right)\; \mathcal{N}\left(\boldsymbol{X}^{(i_{\mathcal{O}})}|\; \boldsymbol{g}\left(\boldsymbol{z}^{(i_{\mathcal{O}})};\boldsymbol{\theta}_g\right),\boldsymbol{S}_X\right) \\
&\qquad \mathcal{N}\left(\boldsymbol{x}^{(i_{\mathcal{O}})}|\; \boldsymbol{f}\left(\boldsymbol{z}^{(i_{\mathcal{O}})};\boldsymbol{\theta}_x\right),\; \boldsymbol{S}_x\left(\boldsymbol{z}^{(i_{\mathcal{O}})};\boldsymbol{\theta}_x\right)\right)\; \mathcal{N}\left(\boldsymbol{z}^{(i_{\mathcal{O}})}|\; \boldsymbol{0},\boldsymbol{I}\right)\; d\boldsymbol{y}^{(i_{\mathcal{O}})}\; d\boldsymbol{X}^{(i_{\mathcal{O}})}\; d\boldsymbol{z}^{(i_{\mathcal{O}})}
\end{aligned}\;,
\tag{23}
$$

where $p(\hat{\boldsymbol{o}}^{(i_{\mathcal{O}})}|\boldsymbol{y}^{(i_{\mathcal{O}})},\boldsymbol{x}^{(i_{\mathcal{O}})};\boldsymbol{\tau})$ depends on the nature of the virtual observable (e.g. Equation (8) or Equation (12)). A fully Bayesian model could be defined by the introduction of appropriate priors for $\boldsymbol{\theta}$ leading to a posterior on those, i.e. $p\left(\boldsymbol{\theta}|\mathcal{D}\right) \propto p\left(\mathcal{D}|\boldsymbol{\theta}\right) p\left(\boldsymbol{\theta}\right)$.

2.5. Inference and learning

Our primary objective is to learn the model parameters $\boldsymbol{\theta}$ on the basis of the mixed data $\mathcal{D} = \{\mathcal{D}_u, \mathcal{D}_s, \mathcal{D}_{\mathcal{O}}\}$ so that the trained probabilistic surrogate can be used for predictive purposes. This task is hindered by the intractability of all the likelihood terms in Equations (21)-(23) due to the presence of the latent variables which must be integrated out. In the following we will discuss how such an intractable model can be trained, even if the likelihood cannot be evaluated in closed form. In order to simplify notation for our following discussion, let us denote summarily by $\mathcal{R} = \{\mathcal{Z}_u, \mathcal{Z}_l, \mathcal{Z}_{\mathcal{O}}, \mathfrak{X}_l, \mathfrak{X}_{\mathcal{O}}, \mathcal{Y}_{\mathcal{O}}\}$ the latent variables appearing in Equations (21) - (23) which consist of:

- $\mathcal{Z}_u = \{\boldsymbol{z}^{(i_u)}\}_{i_u=1}^{N_u}$ associated with $\mathcal{D}_u$ (see, e.g., Equation (4) or Equation (21)),
- $\mathcal{Z}_l = \{\boldsymbol{z}^{(i_l)}\}_{i_l=1}^{N_l}$, $\mathfrak{X}_l = \{\boldsymbol{X}^{(i_l)}\}_{i_l=1}^{N_l}$ associated with $\mathcal{D}_l$ (see, e.g., Equation (3) or (22)),
- $\mathcal{Z}_{\mathcal{O}} = \{\boldsymbol{z}^{(i_{\mathcal{O}})}\}_{i_{\mathcal{O}}=1}^{N_{\mathcal{O}}}$, $\mathfrak{X}_{\mathcal{O}} = \{\boldsymbol{X}^{(i_{\mathcal{O}})}\}_{i_{\mathcal{O}}=1}^{N_{\mathcal{O}}}$, $\mathcal{Y}_{\mathcal{O}} = \{\boldsymbol{y}^{(i_{\mathcal{O}})}\}_{i_{\mathcal{O}}=1}^{N_{\mathcal{O}}}$ associated with $\mathcal{D}_{\mathcal{O}}$ (see, e.g., Equation (23)).

To enable the training of the intractable latent variable model, we advocate the use of Stochastic Variational Inference (SVI, [67,68]), which produces closed-form approximations of the true posterior $p(\boldsymbol{\theta},\mathcal{R}|\mathcal{D})$ and simultaneously of the model evidence $p(\mathcal{D})$. In contrast to sampling-based procedures (e.g., MCMC, SMC), stochastic variational inference yields biased estimates at the benefit of computational efficiency and computable convergence objectives in the form of the Evidence Lower Bound (ELBO [69]). In particular, we denote the variational approximation to the joint posterior as $q_{\boldsymbol{\xi}}(\boldsymbol{\theta},\mathcal{R})$ where $\boldsymbol{\xi}$ are its tunable parameters and note that the model evidence $p(\mathcal{D})$ can be lower-bounded as [70]:

$$
\begin{aligned}
\log p(\mathcal{D}) &= \log \int p(\mathcal{D},\boldsymbol{\theta},\mathcal{R})\, \mathrm{d}\boldsymbol{\theta}\, \mathrm{d}\mathcal{R} \\
&= \mathcal{F}\left(\boldsymbol{\xi}\right) + KL\left(q_{\boldsymbol{\xi}}\left(\boldsymbol{\theta},\mathcal{R}\right)\; ||\; p\left(\boldsymbol{\theta},\mathcal{R}|\mathcal{D}\right)\right)\;, \\
&\geq \mathcal{F}(\boldsymbol{\xi})
\end{aligned}
\tag{24}
$$

where

$$
0 \leq KL\left(q_{\boldsymbol{\xi}}\left(\boldsymbol{\theta},\mathcal{R}\right)\; ||\; p\left(\boldsymbol{\theta},\mathcal{R}|\mathcal{D}\right)\right) = -\int q_{\boldsymbol{\xi}}\left(\boldsymbol{\theta},\mathcal{R}\right) \log\left(\frac{p(\boldsymbol{\theta},\mathcal{R}|\mathcal{D})}{q_{\boldsymbol{\xi}}(\boldsymbol{\theta},\mathcal{R})}\right)\, d\boldsymbol{\theta}\, d\mathcal{R}
\tag{25}
$$

is the KL-divergence between approximate and true posterior, and $\mathcal{F}(\boldsymbol{\xi})$ is the ELBO, i.e.

$$
\begin{aligned}
\mathcal{F}(\boldsymbol{\xi}) &= \int q_{\boldsymbol{\xi}}\left(\boldsymbol{\theta},\mathcal{R}\right)\; \log\left(\frac{p(\mathcal{D},\boldsymbol{\theta},\mathcal{R})}{q_{\boldsymbol{\xi}}(\boldsymbol{\theta},\mathcal{R})}\right)\, d\boldsymbol{\theta}\, d\mathcal{R} \\
&= \mathbb{E}_{q_{\boldsymbol{\xi}}}\left[\log\left(\frac{p(\mathcal{D},\boldsymbol{\theta},\mathcal{R})}{q_{\boldsymbol{\xi}}(\boldsymbol{\theta},\mathcal{R})}\right)\right]
\end{aligned}\;.
\tag{26}
$$

Maximizing the ELBO over the parameters $\boldsymbol{\xi}$ is therefore equivalent to minimizing the KL-divergence from the true posterior. The ELBO provides a score function for comparing different approximations (e.g. different family of distributions $q \in \mathcal{Q}$ or different parametrizations $\boldsymbol{\xi}$) and as an approximation to the model evidence can also be used to compare different models (e.g., with different structure or different parametrizations $\boldsymbol{\theta}$).

We employ a (partial) mean field approximation, i.e. a q_ξ that factorizes as follows

$$q_\xi(\boldsymbol{\theta}, \mathcal{R}) = q_\xi(\boldsymbol{\theta}) \prod_{i_u=1}^{N_u} q_\xi\left(\boldsymbol{z}^{(i_u)}\right) \prod_{i_l=1}^{N_l} q_\xi\left(\boldsymbol{z}^{(i_l)}\right) q_\xi\left(\boldsymbol{X}^{(i_l)}\right) \prod_{i_\mathcal{O}}^{N_\mathcal{O}} q_\xi\left(\boldsymbol{z}^{(i_\mathcal{O})}\right) q_\xi\left(\boldsymbol{X}^{(i_\mathcal{O})}\right) q_\xi\left(\boldsymbol{y}^{(i_\mathcal{O})}\right) . \tag{27}$$

While this might appear drastic, we note that the elements of $\mathcal{Z}_u$ are conditionally (given $\boldsymbol{\theta}$) independent of the rest even in the true posterior. The same holds for the latent variables in the following two groups $\{\mathcal{Z}_l, \mathfrak{X}_l\}$ and $\{\mathcal{Z}_\mathcal{O}, \mathfrak{X}_\mathcal{O}, \mathcal{Y}_\mathcal{O}\}$. Furthermore, $q(\mathcal{R})$ is only an auxiliary distribution which facilitates the training of the intractable generative model (i.e. it only has an impact on later predictions to the extent that it influences $q_\xi(\boldsymbol{\theta})$). Given this, the ELBO becomes:

$$\begin{aligned}
\mathcal{F}(\boldsymbol{\xi}) &= \mathbb{E}_{q_\xi}\left[\log\left(\frac{p(\mathcal{D}, \boldsymbol{\theta}, \mathcal{R})}{q_\xi(\boldsymbol{\theta}, \mathcal{R})}\right)\right] \\
&= \mathbb{E}_{q_\xi}\left[\log p(\mathcal{D}_u|\boldsymbol{\theta}, \mathcal{R}) + \log p(\mathcal{D}_l|\boldsymbol{\theta}, \mathcal{R}) + \log p(\mathcal{D}_\mathcal{O}|\boldsymbol{\theta}, \mathcal{R}) + \log p(\mathcal{R}, \boldsymbol{\theta}) - \log q_\xi(\boldsymbol{\theta}, \mathcal{R})\right] \\
&= \left.\begin{aligned}
&\sum_{i_u=1}^{N_u} \mathbb{E}_{q_\xi}\left[\log p\left(\boldsymbol{x}^{(i_u)}\middle|\boldsymbol{z}^{(i_u)}, \boldsymbol{\theta}\right)\right] \\
&+ \sum_{i_l=1}^{N_l} \mathbb{E}_{q_\xi}\left[\log p\left(\boldsymbol{y}^{(i_l)}\middle|\boldsymbol{X}^{(i_l)}, \boldsymbol{\theta}\right) + \log p\left(\boldsymbol{x}^{(i_l)}\middle|\boldsymbol{z}^{(i_l)}, \boldsymbol{\theta}\right)\right] \\
&+ \sum_{i_\mathcal{O}=1}^{N_\mathcal{O}} \mathbb{E}_{q_\xi}\left[\log p\left(\hat{\boldsymbol{o}}^{(i_\mathcal{O})}\middle|\boldsymbol{y}^{(i_\mathcal{O})}, \boldsymbol{x}^{(i_\mathcal{O})}, \boldsymbol{\theta}\right) + \log p\left(\boldsymbol{x}^{(i_\mathcal{O})}\middle|\boldsymbol{z}^{(i_\mathcal{O})}, \boldsymbol{\theta}\right)\right]
\end{aligned}\right\} \begin{aligned}
&\mathbb{E}_{q_\xi}[\log p(\mathcal{D}_u|\boldsymbol{\theta}, \mathcal{R})] \\
&\mathbb{E}_{q_\xi}[\log p(\mathcal{D}_l|\boldsymbol{\theta}, \mathcal{R})] \\
&\mathbb{E}_{q_\xi}[\log p(\mathcal{D}_\mathcal{O}|\boldsymbol{\theta}, \mathcal{R})]
\end{aligned} \\
&\quad \left.\begin{aligned}
&+ \sum_{i_u=1}^{N_u} \mathbb{E}_{q_\xi}\left[\log p\left(\boldsymbol{z}^{(i_u)}\right)\right] \\
&+ \sum_{i_l=1}^{N_l} \mathbb{E}_{q_\xi}\left[\log p\left(\boldsymbol{X}^{(i_l)}\middle|\boldsymbol{z}^{(i_l)}, \boldsymbol{\theta}\right) + \log p\left(\boldsymbol{z}^{(i)}\right)\right] \\
&+ \sum_{i_\mathcal{O}=1}^{N_\mathcal{O}} \mathbb{E}_{q_\xi}\left[\log p\left(\boldsymbol{y}^{(i_\mathcal{O})}\middle|\boldsymbol{X}^{(i_\mathcal{O})}, \boldsymbol{\theta}\right) + \log p\left(\boldsymbol{X}^{(i_\mathcal{O})}\middle|\boldsymbol{z}^{(i_\mathcal{O})}, \boldsymbol{\theta}\right) + \log p\left(\boldsymbol{z}^{(i_\mathcal{O})}\right)\right]
\end{aligned}\right\} \mathbb{E}_{q_\xi}[\log p(\mathcal{R}|\boldsymbol{\theta})] \\
&\quad + \mathbb{E}_{q_\xi}[\log p(\boldsymbol{\theta})] \\
&\quad - \mathbb{E}_{q_\xi}\left[\log q_\xi(\mathcal{R}) + \log q_\xi(\boldsymbol{\theta})\right].
\end{aligned} \tag{28}$$

In all subsequent illustrations we used point estimates for the parameters $\boldsymbol{\theta}$, i.e. computed their maximum-a-posteriori (MAP) estimate $\boldsymbol{\theta}_{MAP}$. This is equivalent to introducing a Dirac-delta

$$q_\xi(\boldsymbol{\theta}) = \delta(\boldsymbol{\theta} - \boldsymbol{\theta}_{MAP}) \tag{29}$$

in the variational approximation in which case the parameters $\boldsymbol{\xi}$ include also $\boldsymbol{\theta}_{MAP}$. In this case, the expectations with respect to $q_\xi(\boldsymbol{\theta})$ can simply be computed by substituting $\boldsymbol{\theta}_{MAP}$ wherever $\boldsymbol{\theta}$ appears and the entropy term $\mathbb{E}_{q_\xi}[\log q_\xi(\boldsymbol{\theta})]$ can be ignored as it is independent of θ_{MAP}.

The presence of three sets of conditionally independent datasets, i.e. $\mathcal{D}_u$,$\mathcal{D}_l$ and $\mathcal{D}_\mathcal{O}$ (Equation (20)) leads to an additive decomposition of the ELBO of the form $\mathcal{F} = \mathcal{F}_u + \mathcal{F}_l + \mathcal{F}_\mathcal{O} + \log p(\boldsymbol{\theta}_{MAP})$, where

$$\mathcal{F}_u(\boldsymbol{\xi}) = \sum_{i_u=1}^{N_u} \mathbb{E}_{q_\xi}\left[\log p(\boldsymbol{x}^{(i_u)}|\boldsymbol{z}^{(i_u)}, \boldsymbol{\theta})\right] + \sum_{i_u=1}^{N_u} \mathbb{E}_{q_\xi}\left[\log p(\boldsymbol{z}^{(i_u)})\right] - \sum_{i_u=1}^{N_u} \mathbb{E}_{q_\xi}\left[\log q_\xi(\boldsymbol{z}^{(i_u)})\right] \tag{30}$$

accounts for the terms associated with the unlabeled data $\mathcal{D}_u$,

$$\begin{aligned}
\mathcal{F}_l(\boldsymbol{\xi}) &= \sum_{i_l=1}^{N_l} \mathbb{E}_{q_\xi}\left[\log p(\boldsymbol{y}^{(i_l)}|\boldsymbol{X}^{(i_l)}, \boldsymbol{\theta}) + \log p(\boldsymbol{x}^{(i_l)}|\boldsymbol{z}^{(i_l)}, \boldsymbol{\theta})\right] \\
&+ \sum_{i_l=1}^{N_l} \mathbb{E}_{q_\xi}\left[\log p(\boldsymbol{X}^{(i_l)}|\boldsymbol{z}^{(i_l)}, \boldsymbol{\theta}) + \log p(\boldsymbol{z}^{(i_l)})\right] \\
&- \sum_{i_l=1}^{N_l} \mathbb{E}_{q_\xi}\left[\log q_\xi(\boldsymbol{X}^{(i_l)}) + \log q_\xi(\boldsymbol{z}^{(i_l)})\right]
\end{aligned} \tag{31}$$

accounts for the terms associated with the labeled data $\mathcal{D}_l$, and

$$\begin{aligned}
\mathcal{F}_\mathcal{O}(\boldsymbol{\xi}) &= \sum_{i_\mathcal{O}=1}^{N_\mathcal{O}} \mathbb{E}_{q_\xi}\left[\log p(\hat{\boldsymbol{o}}^{(i_\mathcal{O})}|\boldsymbol{y}^{(i_\mathcal{O})}, \boldsymbol{x}^{(i_\mathcal{O})}, \boldsymbol{\theta}) + \log p(\boldsymbol{x}^{(i_\mathcal{O})}|\boldsymbol{z}^{(i_\mathcal{O})}, \boldsymbol{\theta})\right] \\
&+ \sum_{i_\mathcal{O}=1}^{N_\mathcal{O}} \mathbb{E}_{q_\xi}\left[\log p(\boldsymbol{y}^{(i_\mathcal{O})}|\boldsymbol{X}^{(i_\mathcal{O})}, \boldsymbol{\theta}) + \log p(\boldsymbol{X}^{(i_\mathcal{O})}|\boldsymbol{z}^{(i_\mathcal{O})}, \boldsymbol{\theta}) + \log p(\boldsymbol{z}^{(i_\mathcal{O})})\right] \\
&- \sum_{i_\mathcal{O}=1}^{N_\mathcal{O}} \mathbb{E}_{q_\xi}\left[\log q_\xi(\boldsymbol{y}^{(i_\mathcal{O})}) + \log q_\xi(\boldsymbol{X}^{(i_\mathcal{O})}) + \log q_\xi(\boldsymbol{z}^{(i_\mathcal{O})})\right]
\end{aligned} \tag{32}$$

accounts for the terms associated with the virtual observables/data $\mathcal{D}_\mathcal{O}$.

We note that in Equation (30), Equation (31) and Equation (32) the expected log-likelihood terms (i.e. first sum) promote a good fit of the generative model to the unlabeled $\mathcal{D}_u$, labeled $\mathcal{D}_l$ and virtual data $\mathcal{D}_\mathcal{O}$ data respectively, while the second and third sums correspond to the Kullback-Leibler divergence between approximate posteriors and priors which act as regularization that prevents overfitting. The common model parameters $\boldsymbol{\theta}$ appear in all components of the ELBO and synthesize

Algorithm 1: Training generative model using SVI.

Data: Generative Model, $\mathcal{D}_u = \{\boldsymbol{x}^{(i_u)}\}_{i_u=1}^{N_u}$, $\mathcal{D}_l = \{\boldsymbol{x}^{(i_l)}, \boldsymbol{y}^{(i_l)}\}_{i_l=1}^{N_l}$, $\mathcal{D}_{\mathcal{O}} = \left\{\boldsymbol{x}^{(i_{\mathcal{O}})}, \hat{\boldsymbol{o}}^{(i_{\mathcal{O}})}\right\}_{i_{\mathcal{O}}=1}^{N_{\mathcal{O}}}$

while *ELBO not converged* **do**

  // Reparametrization trick

  Sample $\boldsymbol{\epsilon}_{(k)} \sim p(\boldsymbol{\epsilon})$, $k = 1, ..., K$;

  $\mathcal{R}_{(k)} \leftarrow \varrho_{\xi}^{\mathcal{R}}(\boldsymbol{\epsilon}_{(k)})$ $\boldsymbol{\theta}_{(k)} \leftarrow \varrho_{\xi}^{\boldsymbol{\theta}}(\boldsymbol{\epsilon}_{(k)})$ $k = 1, ..., K$;

  // Monte Carlo estimate of ELBO

  Estimate $\hat{\mathcal{F}} \leftarrow \sum_{k=1}^{K} \mathcal{F}(\boldsymbol{\theta}_{(k)}, \mathcal{R}_{(k)})$; // Equation (28)

  // Backpropagate

  $\boldsymbol{g}_{\xi} \leftarrow \nabla_{\xi} \sum_{k=1}^{K} \mathcal{F}(\boldsymbol{\theta}_{(k)}, \mathcal{R}_{(k)})$;

  // Stochastic Gradient Update

  $\boldsymbol{\xi}^{(n+1)} \leftarrow \boldsymbol{\xi}^{(n)} + \boldsymbol{\rho}^{(n)} \odot \boldsymbol{g}_{\xi}$;

  $n \leftarrow n + 1$

end

the information provided by the different data-types. We highlight the term $\log p(\hat{\boldsymbol{o}}^{(i_{\mathcal{O}})} | \boldsymbol{y}^{(i_{\mathcal{O}})}, \boldsymbol{x}^{(i_{\mathcal{O}})}, \boldsymbol{\theta})$ in Equation (32), which is driven by the virtual dataset and reflects the incorporation of our (in)equality constraints. In this case, the model attempts to infer the solution $\boldsymbol{y}^{(i_{\mathcal{O}})}$ through $q_{\xi}\left(\boldsymbol{y}^{(i_{\mathcal{O}})}\right)$. Hence the updates of the model parameters $\boldsymbol{\theta}$ are affected also by the inferred solutions and the uncertainty associated with them.

For the structured mean-field approximation $q_{\xi}(\boldsymbol{\theta}, \mathcal{R})$ in Equation (27) we adopt diagonal Gaussians, primarily due to their linear scaling with the dimension of the corresponding latent variables. The following forms and parametrizations for the variational posteriors q_{ξ} in Equation (27) were adopted:

- $\forall i_u \in \{1, ..., N_u\}$: $q_{\xi}\left(\boldsymbol{z}^{(i_u)}\right) = \mathcal{N}\left(\boldsymbol{z}^{(i_l)} \,\middle|\, \boldsymbol{\mu}_{\boldsymbol{z}}^{(i_u)}, \mathrm{diag}\left(\boldsymbol{\sigma}_{\boldsymbol{z}}^{(i_u)}\right)\right)$
- $\forall i_l \in \{1, ..., N_l\}$: $q_{\xi}\left(\boldsymbol{z}^{(i_u)}\right) = \mathcal{N}\left(\boldsymbol{z}^{(i_l)} \,\middle|\, \boldsymbol{\mu}_{\boldsymbol{z}}^{(i_l)}, \mathrm{diag}\left(\boldsymbol{\sigma}_{\boldsymbol{z}}^{(i_l)}\right)\right)$ $q_{\xi}\left(\boldsymbol{X}^{(i_l)}\right) = \mathcal{N}\left(\boldsymbol{X}^{(i_l)} \,\middle|\, \boldsymbol{\mu}_{\boldsymbol{X}}^{(i_l)}, \mathrm{diag}\left(\boldsymbol{\sigma}_{\boldsymbol{X}}^{(i_l)}\right)\right)$
- $\forall i_{\mathcal{O}} \in \{1, ..., N_{\mathcal{O}}\}$: $q_{\xi}\left(\boldsymbol{z}^{(i_{\mathcal{O}})}\right) = \mathcal{N}\left(\boldsymbol{z}^{(i_{\mathcal{O}})} \,\middle|\, \boldsymbol{\mu}_{\boldsymbol{z}}^{(i_{\mathcal{O}})}, \mathrm{diag}\left(\boldsymbol{\sigma}_{\boldsymbol{z}}^{(i_{\mathcal{O}})}\right)\right)$ $q_{\xi}\left(\boldsymbol{X}^{(i_{\mathcal{O}})}\right) = \mathcal{N}\left(\boldsymbol{X}^{(i_{\mathcal{O}})} \,\middle|\, \boldsymbol{\mu}_{\boldsymbol{X}}^{(i_{\mathcal{O}})}, \mathrm{diag}\left(\boldsymbol{\sigma}_{\boldsymbol{X}}^{(i_{\mathcal{O}})}\right)\right)$
  $q_{\xi}\left(\boldsymbol{y}^{(i_{\mathcal{O}})}\right) = \mathcal{N}\left(\boldsymbol{y}^{(i_{\mathcal{O}})} \,\middle|\, \boldsymbol{\mu}_{\boldsymbol{y}}^{(i_{\mathcal{O}})}, \mathrm{diag}\left(\boldsymbol{\sigma}_{\boldsymbol{y}}^{(i_{\mathcal{O}})}\right)\right)$

which, in combination with Equation (29) suggest that the parameter vector $\boldsymbol{\xi}$ consists of

$$\boldsymbol{\xi} = \left\{\boldsymbol{\theta}_{MAP},\ \left\{\boldsymbol{\mu}_{\boldsymbol{z}}^{(i_u)}, \boldsymbol{\sigma}_{\boldsymbol{z}}^{(i_u)}\right\}_{i_u=1}^{N_u} \left\{\boldsymbol{\mu}_{\boldsymbol{z}}^{(i_l)}, \boldsymbol{\sigma}_{\boldsymbol{z}}^{(i_l)}, \boldsymbol{\mu}_{\boldsymbol{X}}^{(i_l)}, \boldsymbol{\sigma}_{\boldsymbol{X}}^{(i_l)}\right\}_{i_l=1}^{N_l} \left\{\boldsymbol{\mu}_{\boldsymbol{z}}^{(i_{\mathcal{O}})}, \boldsymbol{\sigma}_{\boldsymbol{z}}^{(i_{\mathcal{O}})}, \boldsymbol{\mu}_{\boldsymbol{X}}^{(i_{\mathcal{O}})}, \boldsymbol{\sigma}_{\boldsymbol{X}}^{(i_{\mathcal{O}})}, \boldsymbol{\mu}_{\boldsymbol{y}}^{(i_{\mathcal{O}})}, \boldsymbol{\sigma}_{\boldsymbol{y}}^{(i_{\mathcal{O}})}\right\}_{i_{\mathcal{O}}=1}^{N_{\mathcal{O}}}\right\}. \quad (33)$$

For the parameters that are constrained to be positive, a suitable transformation (e.g. $\exp(\cdot)$) is employed such that maximizing the ELBO becomes an unconstrained optimization problem.[6]

From Equation (33) it is obvious that the number of variational parameters associated with the, potentially large unlabeled dataset, $\mathcal{D}_u$ scales linearly with N_u. One may therefore consider introducing an *amortized* encoder $q_{\Phi}\left(\boldsymbol{z}^{(i_u)} \middle| \boldsymbol{x}^{(i_u)}\right)$ [71], i.e. an approximate posterior that explicitly accounts for the dependence of each $\boldsymbol{z}^{(i_u)}$ on the data $\boldsymbol{x}^{(i_u)}$. In particular, we adopt an approximate posterior of the form

$$q_{\Phi}\left(\boldsymbol{z}^{(i_u)} \middle| \boldsymbol{x}^{(i_u)}\right) = \mathcal{N}\left(\boldsymbol{z}^{(i_u)} \middle| \boldsymbol{\mu}_{\Phi}\left(\boldsymbol{x}^{(i_u)}\right), \mathrm{diag}\left(\boldsymbol{\sigma}_{\Phi}\left(\boldsymbol{x}^{(i_u)}\right)\right)\right) \qquad \forall i_u \in \{1, ..., N_u\}\,, \quad (34)$$

where the amortization implies that the parameters Φ are shared between all instances i_u of unlabeled data. Similarly to the choice of $q(\mathcal{R})$ the specific structure of Eq. (34) follows from numerical considerations.[7] While the approximate posterior in Equation (34) can, at best, achieve the same ELBO as the $q_{\xi}(\boldsymbol{z}^{(i_u)})$ above, it contains fewer parameters that need to be optimized (at least for large N_u) and once trained can be readily used as an approximation to the true posterior $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ for predictive purposes in Equation (15). In our simulations, the parameters Φ pertain to deep neural nets (see section 3) and from a practical point of view, the only difference is that $\left\{\boldsymbol{\mu}_{\boldsymbol{z}}^{(i_u)}, \boldsymbol{\sigma}_{\boldsymbol{z}}^{(i_u)}\right\}_{i=1}^{N_u}$ are substituted by the parameters Φ in the vector $\boldsymbol{\xi}$ of Equation (33), and that the unlabeled data is subsampled in batches during training.

We conclude this section by enumerating the basic steps associated with the variational inference task in Algorithm 1. The intractable expectations with respect to q_{ξ} appearing in the ELBO $\mathcal{F}$ and its gradient $\nabla_{\xi}\mathcal{F}$ are estimated with Monte Carlo. In order to reduce the variance of these estimators, we apply the well-established reparametrization trick [71].

[6] We note that $\boldsymbol{\sigma}$ denotes a vector of *variances*, not standard deviations.

[7] This specific choice is amenable to *reparametrization* (see Algorithm 1). As detailed in the seminal paper of [71] this enables low-variance estimates of the gradients of the ELBO needed in training (see Algorithm 2).

Algorithm 2: Making predictions for new $\boldsymbol{x}$ using the generative model.

Data: $\boldsymbol{x}$, trained generative model
if *amortization* **then**
  $q^*(\boldsymbol{z}) \leftarrow q_\Phi(\boldsymbol{z}|\boldsymbol{x})$; // Equation (34)
else
  $q^*(\boldsymbol{z}) \leftarrow \arg\max_\zeta \hat{\mathcal{F}}_u(q_\zeta(\boldsymbol{z}))$; // Equation (37)
end
for $k \leftarrow 1$ **to** K **do**
  Sample $\boldsymbol{z}^{(k)} \sim q^*(\boldsymbol{z})$;
  Sample $\boldsymbol{X}^{(k)} \sim p(\boldsymbol{X}|\boldsymbol{z}^{(k)}, \boldsymbol{\theta}_{MAP})$; // Equation (18)
  Sample $\boldsymbol{y}^{(k)} \sim p(\boldsymbol{y}|\boldsymbol{X}^{(k)}, \boldsymbol{\theta}_{MAP})$; // Equation (19)
end
Construct sample-based approximation $\tilde{p}(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D})$ using samples $\boldsymbol{y}^{(k)}, k = 1, \ldots, K$

We combine the noisy estimates of the gradient $\nabla_{\boldsymbol{\xi}}\mathcal{F}$ with stochastic gradient ascent [72] and the Adam algorithm in particular [73]. We note that training requires the propagation of gradients through the whole model, including the CGM and the constraints associated with virtual observables. Propagating gradients through the model can readily be done using algorithmic differentiation [74] whenever possible; i.e., when evaluating a Monte Carlo estimate of the evidence lower bound $\mathcal{F}$ a computational graph is built, such that in a backward pass gradient information propagates from $\mathcal{F}$ to the leaf nodes of the computational graph (e.g. given by the variational parameters $\boldsymbol{\xi}$) [75]. The CGM and the virtual observables $\boldsymbol{o}(\boldsymbol{y}; \boldsymbol{x})$ must be embedded within this computational graph, i.e. it is required that the CGM also allows the back-propagation of gradient information. If the CGM involves the solution of a (coarse-grained) PDE, the reverse-flow of information required during back-propagation corresponds to the solution of the adjoint problem (at a cost equivalent to the forward solution of the CGM). Obtaining derivatives of the virtual observables is equally a cheap operation but also problem-specific and discussion is deferred until section 3.3.

2.6. Predictions

Once an (approximate) posterior $q_{\boldsymbol{\xi}}(\boldsymbol{\theta})$ on the model parameters $\boldsymbol{\theta}$ has been computed, the interest shifts to using the trained model for predictions. The adoption of a generative model however implies that by learning a joint distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{y})$, the desired posterior predictive $p(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D})$ no longer directly exists in closed form. In the simplest case, given a new (unobserved) input $\boldsymbol{x}$, we seek the corresponding output $\boldsymbol{y}$. The probabilistic nature of the proposed generative model yields a probability density on $\boldsymbol{y}$ (see also (5)), i.e. the predictive posterior $p(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D})$ given by

$$p(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D}) = \int p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta})\, p(\boldsymbol{X}|\boldsymbol{z}, \boldsymbol{\theta})\, p(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta})\, p(\boldsymbol{\theta}|\mathcal{D})\; d\boldsymbol{z}\, d\boldsymbol{X}\, d\boldsymbol{\theta} \tag{35}$$

$$\approx \int p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}_{MAP})\, p(\boldsymbol{X}|\boldsymbol{z}, \boldsymbol{\theta}_{MAP})\, p(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}_{MAP})\; d\boldsymbol{X}\, d\boldsymbol{z}\,, \tag{36}$$

where the variational approximation $q_{\boldsymbol{\xi}}(\boldsymbol{\theta}) = \delta(\boldsymbol{\theta} - \theta_{MAP})$ was used in place of the intractable posterior $p(\boldsymbol{\theta}|\mathcal{D})$.[8]

If an amortized approximate posterior $q_\Phi(\boldsymbol{z}|\boldsymbol{x})$ has been found in the inference step as detailed in the previous section, then this can be used in place of $p(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}_{MAP})$ in Equation (36). Alternatively, one might employ sampling methods (e.g. MCMC) or another round of (stochastic) variational inference in order to obtain an approximation, say $q_\zeta(\boldsymbol{z})$. The latter is found by maximizing an analogous ELBO, i.e.

$$\begin{aligned} q^*(\boldsymbol{z}) &= \arg\min_\zeta \mathrm{KL}\left[q_\zeta(\boldsymbol{z}) \middle|\middle| p(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}_{MAP})\right] \\ &= \arg\max_\zeta \mathbb{E}_{q_\zeta(\boldsymbol{z})}\left[\log p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\theta}_{MAP})\right] - \mathrm{KL}\left[q_\zeta(\boldsymbol{z}) \middle|\middle| p(\boldsymbol{z})\right] \\ &= \arg\max_\zeta \hat{\mathcal{F}}_u\left(q_\zeta(\boldsymbol{z})\right)\,. \end{aligned} \tag{37}$$

We note that irrespective of the adopted method, no additional model solves of the FGM are required and for the results reported in subsequent sections the variational approximation q_ζ was used. The integral in the predictive posterior of (36) can be approximated with Monte Carlo and requires solely solutions of the CGM. In Algorithm 2 we briefly summarize how probabilistic predictions $p(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D})$ can be obtained for new (unobserved) inputs $\boldsymbol{x}$.

[8] We also briefly mention the possibility (without pursuing it further in this work) to incorporate (additional) constraints $\boldsymbol{o}(\boldsymbol{y}; \boldsymbol{x})$ at $\boldsymbol{x}$ during the prediction stage as well, i.e. to perform *prediction by inference* and update the posterior predictive using again the *virtual likelihood* $p(\boldsymbol{y}|\boldsymbol{x}, \hat{\boldsymbol{o}}, \mathcal{D}) \propto p(\hat{\boldsymbol{o}}|\boldsymbol{y}, \boldsymbol{x})\, p(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D})$ where $\hat{\boldsymbol{o}}$ denotes the associated virtual observables.

2.6.1. Predictive performance metrics

In the context of making (probabilistic) predictions, it is essential to score the predictive utility of the probabilistic surrogate in a way that assesses how well the model has learned to generalize the underlying mapping (i.e. the mapping $\boldsymbol{y}(\boldsymbol{x})$ implicitly defined by the PDE and the FGM). To this end we consider a validation dataset $\mathcal{D}_v = \left\{\boldsymbol{x}^{(i_v)}, \boldsymbol{y}^{(i_v)}\right\}_{i_v=1}^{N_v}$ consisting of N_v input-output pairs of the FGM *not appearing in the training data*. On this validation dataset we evaluate the following two metrics using the predictive posterior density:

Coefficient of determination R^2 The coefficient of determination R^2 is a standard metric [76] which assesses the accuracy of point estimates, and in particular of the mean $\boldsymbol{\mu}(\boldsymbol{x}^{(i_v)})$ of the predictive posterior of our trained model for each validation input $\boldsymbol{x}^{(i_v)}$, i.e.

$$\boldsymbol{\mu}(\boldsymbol{x}^{(i_v)}) = \mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x}^{(i_v)},\mathcal{D})}[\boldsymbol{y}], \qquad i_v = 1, \dots, N_v\,. \tag{38}$$

The mean of the posterior predictive is estimated using Monte Carlo (see Algorithm 2) and is compared to the reference FGM outputs $\{\boldsymbol{y}^{(i_v)}\}_{i=1}^{N_v}$ using the coefficient of determination

$$R^2 = 1 - \frac{\sum_{i_v=1}^{N_v} \left|\left|\boldsymbol{y}^{(i_v)} - \boldsymbol{\mu}(\boldsymbol{x}^{(i_v)})\right|\right|_2^2}{\sum_{i_v=1}^{N_v} \left|\left|\boldsymbol{y}^{(i_v)} - \boldsymbol{y}_v\right|\right|_2^2}, \tag{39}$$

where $\boldsymbol{y}_v = \frac{1}{N_v}\sum_{i_v=1}^{N_v} \boldsymbol{y}^{(i_v)}$ is the sample average of the validation dataset. It can be noted that R^2 attains its maximum value, i.e. $R^2 = 1$, when the mean predictive estimates coincide with the actual FGM outputs in the validation dataset and deviations from these are weighted by the variability of the validation data appearing in the denominator of Equation (39).

Logscore LS This metric assesses not just point estimates of the predictive posterior but also the associated predictive uncertainty. In particular and for the purpose of computing LS we approximate the otherwise intractable $p(\boldsymbol{y}|\boldsymbol{x}^{(i_v)}, \mathcal{D})$ in Equation (36) at each validation input $\boldsymbol{x}^{(i_v)}$, by a Gaussian with a mean equal to the actual mean of the predictive posterior $\boldsymbol{\mu}(\boldsymbol{x}^{(i_v)})$ (Equation (38) - estimated by Monte Carlo) and a diagonal covariance matrix $\boldsymbol{S}(\boldsymbol{x}^{(i_v)})$ containing the actual variances (also estimated by Monte Carlo - see Algorithm 2), i.e.

$$\boldsymbol{S}(\boldsymbol{x}^{(i_v)}) = \mathrm{diag}\left(\sigma_j^2\left(\boldsymbol{x}^{(i_v)}\right)\right), \qquad i_v = 1, \dots, N_v \tag{40}$$

where

$$\sigma_j^2(\boldsymbol{x}^{(i_v)}) = \mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x}^{(i_v)},\mathcal{D})}\left[(y_j - \mu_j(\boldsymbol{x}^{(i_v)}))^2\right], \qquad i_v = 1, \dots, N_v\,. \tag{41}$$

Subsequently, LS is evaluated as

$$LS = \frac{1}{N_v}\sum_{i_v=1}^{N_v} \log\mathcal{N}\left(\boldsymbol{y}^{(i_v)} \mid \boldsymbol{\mu}(\boldsymbol{x}^{(i_v)}), \boldsymbol{S}(\boldsymbol{x}^{(i_v)})\right). \tag{42}$$

One notes that high LS values are achieved not only when the predictive mean $\boldsymbol{\mu}(\boldsymbol{x}^{(i_t)})$ is close to the true $\boldsymbol{y}^{(i_t)}$ but also when the predictive uncertainty (as measured by the variances $\sigma_j^2(\boldsymbol{x}^{(i_t)})$) is simultaneously as small as possible. It can finally be shown [57] that LS approximates the Kullback-Leibler divergence between the true $p(\boldsymbol{y}|\boldsymbol{x})$ and the (Gaussian approximation of the) predictive posterior $p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D})$ averaged over the true distribution, say $p(\boldsymbol{x})$, of the inputs.

2.7. Numerical complexity analysis

In the following we discuss the computational complexity of the proposed algorithms and their scaling with the dimensions of the problem, as well as with the number of, virtual or actual, training data. In such a discussion it is necessary to distinguish between the training phase (i.e., obtaining $\boldsymbol{\theta}_{\mathrm{MAP}}$ - frequently referred to as *offline* phase) and the prediction phase (frequently referred to as *online* phase). Since the CGM is directly embedded in the probabilistic graphical model, the numerical cost of training (with the exception of unlabeled data) depends on the cost of the CGM, which we need to solve for a forward pass of our model (as well as an adjoint solve of the CGM for the backpropagation of gradient information). Forward evaluations of the CGM are also required, if - after training - the model is used for predictive purposes. As such, the overall numerical complexity depends on $d_{cgm} \approx \dim(\boldsymbol{Y}) \approx \dim(\boldsymbol{X})$. The numerical effort of the entire algorithm therefore scales with d_{cgm}, and the specific dependence follows from the type of the CGM; i.e., how the numerical discretization technique used for the CGM scales with the dimension of d_{cgm}. In the following we shall assume $\mathcal{O}(d_{cgm}^2)$ and note that d_{cgm} and the cost of a CGM solve is by construction much smaller than the corresponding dimension and cost of the FGM.

During training the algorithm exhibits linear scaling in the number of labeled data points N_l and query points $N_{\mathcal{O}}$, as variational inference is carried out separately for $q_{\boldsymbol{\xi}}(\boldsymbol{z}^{(i)})$ and $q_{\boldsymbol{\xi}}(\boldsymbol{X}^{(i)})$ for $i = 1, ..., (N_l + N_{\mathcal{O}})$. The same statement extends to the memory requirements resulting from the variational inference for the $\boldsymbol{X}^{(i)}$ and $\boldsymbol{z}^{(i)}$. In contrast, sub-linear scaling can be achieved in terms of the number of unlabeled data N_u, assuming that an amortized encoder $q_\Phi(\boldsymbol{z}|\boldsymbol{x})$ is introduced which enables the batched sub-sampling of data. In addition, the number of parameters $\dim(\Phi)$ of the amortized encoder which one has to infer is constant irrespective of N_u. One of they key points is of course that the virtual observables enable the incorporation of a set of $M = \dim(\hat{\boldsymbol{o}})$ physical constraints at a cost that is dictated by the number of constraints M, and does not directly relate to the dimension arising from the fine-scale discretization, i.e. $d_y = \dim(\boldsymbol{y})$ (in Appendix B we discuss the special case of closed form updates with the complexity being bounded by $\mathcal{O}(M^3)$) . As such the incorporation of virtual observables and the subsequent optimization of $\mathcal{F}_{\mathcal{O}}$ scales overall as $\mathcal{O}(N_{\mathcal{O}} \cdot M^3 \cdot d_{cgm}^2)$.

The cost of the generation of predictive estimates with the trained model is dictated primarily by the cost of the forward solve of the CGM, which makes the surrogate usable in a *multi-query* setting (for which we provide a numerical illustration in section 3.9). Since $p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D})$ is not available in closed form, several evaluations of the CGM (at an assumed complexity $\mathcal{O}(d_{cgm}^2)$ each) are required to obtain a sufficient estimates of the integrals involved (see section 2.6 and Algorithm 2). The numerical cost in the prediction phase is further reduced if an amortized encoder $q_\Phi(\boldsymbol{z}|\boldsymbol{x})$ has been employed, since this enables to bypass variational inference for any new $\boldsymbol{x}$ at which the surrogate is to be evaluated. Hence, FGM solves are needed only in the generation of N_l labeled data $\mathcal{D}_l = \{\boldsymbol{x}^{(i_l)}, \boldsymbol{y}^{(i_l)}\}_{i_l=1}^{N_l}$ provide to the model. Since the cost of each FGM call for most problems outweighs the others, the primary cost metric used for our illustrations is the number of labeled data, which we try to reduce as much as possible while retaining predictive accuracy.

3. Numerical illustrations

We demonstrate the capabilities of the proposed framework in discovering predictive, probabilistic surrogates on a two-dimensional diffusion problem. In the sequel, we specify particular elements of the proposed model that were presented generically in the previous sections and additionally concretize parametrizations and their meaning. The goals of the numerical illustrations are:

- to examine the effect of the number labeled data N_l which are the most expensive to obtain and to assess whether the model can perform well under small N_l (i.e. a few tens of FGM runs, section 3.4).
- to assess the ability of the model to learn effective and interpretable CGMs that provide insight to the relevant features of the high-dimensional input $\boldsymbol{x}$ which are predictive of the output $\boldsymbol{y}$ (section 3.4).
- to examine the effect of the *amount* of virtual observables $\mathcal{D}_{\mathcal{O}}$ and assess whether the model's predictive performance can be improved by increasing the number $N_{\mathcal{O}}$ of such data (section 3.5).
- to examine the effect of the *type* of virtual observables provided for training. In particular, we consider three different types (namely coarse-grained residuals, hybrid and potential energy) and assess the model's predictive performance for each one of those (section 3.5).
- to examine the effect of unlabeled data $\mathcal{D}_u$ which are inexpensive to obtain and to assess whether the model's predictive performance can be improved by increasing the number N_u of such data (section 3.6).
- to examine the effect of the information bottleneck implied by the latent variables $\boldsymbol{z}$ and the CGM and to assess the effect of the dimension of $\boldsymbol{z}$ and the CGM's state variables (i.e. $\boldsymbol{X}$ and $\boldsymbol{Y}$) on the predictive performance of the model (section 3.7)
- to assess the predictive performance of the model under high-dimensional parametric inputs $\boldsymbol{x}$ and under "interpolative" and "extrapolative" conditions. The latter distinction refers to the ability to predict the (equally high-dimensional) output vector $\boldsymbol{y}$ under boundary conditions that were (interpolative) or not (extrapolative) used during training (section 3.8).
- to investigate the efficiency and accuracy of the trained surrogate in a many-query application involving uncertainty propagation (section 3.9).

Some of the simulation results as well as the corresponding code will be made available at the following github repository[9] upon publication.

3.1. Definition of physical problem

For the numerical illustration of our modeling framework we consider a linear elliptic PDE defined on the unit square $\Omega = [0, 1]^d$ in dimension $d = 2$. We can write the governing equations as a two-field problem

$$\text{conservation law: } \nabla \cdot \boldsymbol{J}(\boldsymbol{s}) = f, \quad \forall \boldsymbol{s} \in \Omega \tag{43}$$

$$\text{constitutive law: } \boldsymbol{J}(\boldsymbol{s}) = -\nabla(\kappa(\boldsymbol{s})\, u(\boldsymbol{s})) \quad \forall \boldsymbol{s} \in \Omega \tag{44}$$

[9] https://github.com/bdevl/PGMCPC.

with boundary conditions

$$u = u_D, \quad \boldsymbol{s} \in \Gamma_D \tag{45}$$

$$\boldsymbol{J} \cdot \boldsymbol{n} = \boldsymbol{0}, \quad \boldsymbol{s} \in \Gamma_N\,, \tag{46}$$

where $u(\boldsymbol{s})$ is a scalar field to which one might attribute the physical meaning of temperature or pressure or concentration, $\boldsymbol{J}(\boldsymbol{s})$ is a vector field representing *flux*, and $\boldsymbol{n}$ is the unit outward normal vector. Γ_N denotes the part of the boundary where Neumann boundary conditions are prescribed and is comprised of the top and bottom sides of the unit square Ω, i.e. for $\{\boldsymbol{s}|s_2 = 0 \text{ or } s_2 = 1\}$. At the remaining boundary Γ_D, i.e. the left and right side of the domain, we introduce randomized boundary conditions of the form

$$\begin{aligned} u_D(\boldsymbol{s}) &= a_0 \cdot s_2 + a_1(1 - s_2) \qquad \boldsymbol{s} \in \{\boldsymbol{s}|s_1 = 0\} \\ u_D(\boldsymbol{s}) &= a_2 \cdot s_2 + a_3(1 - s_2) \qquad \boldsymbol{s} \in \{\boldsymbol{s}|s_1 = 1\} \end{aligned} \tag{47}$$

with $a_i \sim \mathcal{U}[-0.5, 0.5]$.
We model $\kappa(\boldsymbol{s})$ with a log-normally distributed random field, i.e., $\kappa(\boldsymbol{s}) = e^{\lambda(\boldsymbol{s})}$ where the underlying Gaussian field has a spatially constant mean μ_λ and a covariance $\mathcal{C}_\lambda(\boldsymbol{s}, \boldsymbol{s}')$ function given by

$$\mathcal{C}_\lambda(\boldsymbol{s}, \boldsymbol{s}') = \sigma_\lambda^2 \cdot \exp\left(-\frac{1}{2}\frac{||\boldsymbol{s} - \boldsymbol{s}'||_2^2}{l_\lambda^2}\right). \tag{48}$$

The following values were used for the parameters: $\mu_\lambda = 0.4$, $\sigma_\lambda = 0.8$ and $l_\lambda = 0.04$ or 0.15 (depending on the resolution of the FGM). The resulting random field $\kappa(\boldsymbol{s})$ exhibits significant variability with a coefficient of variation of 0.95 and the small correlation lengths necessitate fine discretizations resulting in a high-dimensional random input $\boldsymbol{x}$. A discretized sample of $\kappa(\boldsymbol{s})$ is obtained by sampling the underlying Gaussian field on a spatial grid defined by the discretization of the FGM, which will be discussed in the following.

The numerical solution of the governing equations is obtained using a standard Finite Element (FE) schemes. For the purposes of our illustrations we consider the following two FE discretizations giving rise to the fine-grained (FGM) and coarse-grained (CGM) models in the previous discussion:

FGM This employs a fine(r) discretization using a regular grid of size $d_f \times d_f$.[10] Our simulations are based on $d_f = 32$ (for $l_\lambda = 0.15$) and $d_f = 64$ (for $l_\lambda = 0.04$) giving rise to $dim(\boldsymbol{y}) = (d_f + 1)^2$ using the standard FE scheme, i.e. $dim(\boldsymbol{y}) = 1089$ and 4225, respectively. The random field $\kappa(\boldsymbol{s})$ is discretized using piece-wise constant functions over each grid element, and the vector $\boldsymbol{x}$ represents the value of $\kappa(\boldsymbol{s})$ at the centroid of each pixel. Hence $dim(\boldsymbol{x}) = d_f^2$.
In anticipation of the *virtual observables* that will be enforced and are discussed in more detail in section 3.3, we review here the weak form of the governing PDE which, in view of Equation (43) and the boundary conditions in Equation (45) and Equation (46) becomes

$$-\int_\Omega \nabla_s w \cdot \boldsymbol{J}\, d\boldsymbol{s} - \int_\Omega w\, f\, d\boldsymbol{s} = 0\,, \tag{49}$$

or upon making use of the constitutive equation (44)

$$\int_\Omega \nabla_s w \cdot \kappa\, \nabla_s u\, d\boldsymbol{s} - \int_\Omega w\, f\, d\boldsymbol{s} = 0\,. \tag{50}$$

The *admissible* weight functions $w \in \mathcal{W}$ belong in the set $\mathcal{W} = \{w(\boldsymbol{s}) \mid w(\boldsymbol{s}) \in H^1(\Omega), w(\boldsymbol{s}) = 0 \text{ on } \Gamma_D\}$. We denote by $\boldsymbol{y}$ the discretized representation of $u(\boldsymbol{s})$ with the usual FE shape functions which, upon substitution in Equation (50), and for each $w \in \mathcal{W}$ yields a residual $r_w : \mathbb{R}^{d_x} \times \mathbb{R}^{d_y} \to \mathbb{R}$

$$r_w(\boldsymbol{y}; \boldsymbol{x}) = 0\,. \tag{51}$$

We note that depending on the choice of the weight functions w (at least) six methods (i.e. collocation, sub-domain, least-squares, (Petrov)-Galerkin, moments) arise as special cases [79].

[10] The use of regular grids is pursued in order to enable the use of convolutional neural networks (CNNs) ([77], [78]) for the parameterized densities, enabling a parsimonious description of a complex hierarchy of features. We note that expressing physically meaningful spatio-(temporal) features on possibly non-regular and unstructured domains is a challenge in itself, but not the subject of this investigation. As such we have chosen to constrain ourselves to the representation of the random field on a regular grid, which enables the use of methods that have reached maturity due to their extensive use in computer vision.

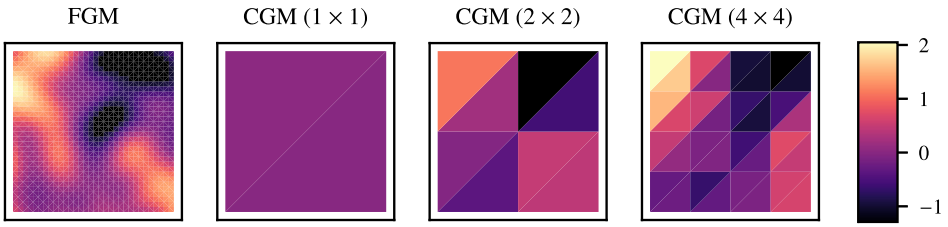


Fig. 4. Comparison of a sample $\boldsymbol{x}^{(i)}$ of the discretized of the Gaussian random field $\lambda(\boldsymbol{s})$ of the FGM (left - Equation (48) with $l_\lambda = 0.15$) with the (log of the posterior mean of the) corresponding $\boldsymbol{X}^{(i)}$ for three different CGM discretizations, i.e. 1×1 , 2×2 and 4×4 (The posterior means $\mathbb{E}[q(\boldsymbol{X}^{(i)})]$ are based on $N_l = 512$ training data). The CGMs encode *effective* properties $\boldsymbol{X}^{(i)}$ via the trained model density $p(\boldsymbol{X}|\boldsymbol{x})$. As the CGM is refined, it captures more details of the underlying FGM properties, e.g. areas in the problem domain with higher/lower conductivity $\boldsymbol{x}$ in the FGM correspond to higher/lower values of $\boldsymbol{X}$ in the CGM. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

It is also well-known that the solution to this problem, as with many problems in computational physics, can be obtained by minimizing an appropriate functional which in this case reduces to the potential energy function $\mathcal{V}$ given by

$$\mathcal{V} = \frac{1}{2}\int_\Omega \kappa\, |\nabla_s u|^2 \; d\boldsymbol{s} - \int_\Omega f\; u\; d\boldsymbol{s}\,. \tag{52}$$

Upon discretization, this suggests that the solution vector $\boldsymbol{y}$ can be found by minimizing V, i.e.

$$min_{\boldsymbol{y}}\; V(\boldsymbol{y};\boldsymbol{x})\,, \tag{53}$$

where V is the discretized potential energy obtained by using the discretized versions of κ and u in $\mathcal{V}$ of Equation (52). We note that the output vector $\boldsymbol{y}$ which corresponds to the discretization of $u(\boldsymbol{s})$ is of similar dimension $d_y = dim(\boldsymbol{y}) = (d_f+1)^2$ as well[11] (Fig. 4). We do not consider the discretization error of the FGM, as our goal in this work is to predict $\boldsymbol{y}$ (i.e. the discretized solution), and as such assume it to be of sufficient accuracy.

CGM This is based on a FE solver on a coarse(r) regular grid of size $d_c \times d_c$. Analogously to the FGM, the CGM input vector $\boldsymbol{X}$ represents the property within each of the pixels and is therefore of dimension $dim(\boldsymbol{X}) = d_c^2$. The FE solver yields the output vector $\boldsymbol{Y}$ (which represents $u(\boldsymbol{s})$) and is therefore of dimension $dim(\boldsymbol{Y}) = (d_c+1)^2$ as well.[12] The values $d_c = 1, 2, 4$ were considered (see Fig. 4) - in all cases $d_c << d_f$) in order to assess the effect of the dimensionality of the CGM in the predictive estimates. We note that this particular form of the CGM was adopted for simplicity and due to the fact that boundary conditions can be readily incorporated in it rather than having to learn their effect as well (e.g. by including them in $\boldsymbol{x}, \boldsymbol{X}$). Nevertheless, any coarse-grained or reduced-order model from the vast literature on this topic can be employed instead.

3.2. Specification of the generative model

Given the physical problem above and the definitions of the associated input $\boldsymbol{X}, \boldsymbol{x}$ and output vectors $\boldsymbol{Y}, \boldsymbol{y}$, we provide details on the parameterization of the generative model which was generically described in section 2. In particular, the following modeling choices were made:

(a) we employ a densely connected convolutional neural network [80] to parameterize the mean $\boldsymbol{f}(\boldsymbol{z};\boldsymbol{\theta}_x)$ as well as the input-dependent diagonal covariance matrix $\boldsymbol{S}_{\boldsymbol{x}}(\boldsymbol{z};\boldsymbol{\theta}_x)$ in Equation (17). In addition, we make use of the same architecture for the amortized encoder $q_\Phi(\boldsymbol{z}|\boldsymbol{x})$ (section 2.5). More specifically, the implementation is based on a variation of the architecture proposed in [34]. The alterations refer predominantly to a reduction in the complexity and expressivity since the latent space $\boldsymbol{z}$ encodes the salient features of $\boldsymbol{x}$, i.e., we primarily wish to retain information to the extent that it can help us in predicting effective properties by means of $p(\boldsymbol{X}|\boldsymbol{z},\boldsymbol{\theta})$ (Equation (18)).

(b) The conditional density $\mathcal{N}(\boldsymbol{X}|\boldsymbol{g}(\boldsymbol{z};\boldsymbol{\theta}_g), \boldsymbol{S}_{\boldsymbol{X}})$ defined by Equation (18) relates the latent encoding $\boldsymbol{z}$ to the input $\boldsymbol{X}$ of the CGM (i.e. the apparent/effective/homogenized properties). The mean vector $\boldsymbol{g}(\boldsymbol{z};\boldsymbol{\theta}_g)$ depends on the latent variables $\boldsymbol{z}$ and is parameterized using a linear layer, i.e. $\boldsymbol{g}(\boldsymbol{z};\boldsymbol{\theta}_g) = \boldsymbol{W}_g\boldsymbol{z} + \boldsymbol{b}_g$ such that $\boldsymbol{\theta}_g = \{\boldsymbol{W}_g, \boldsymbol{b}_g\}$, which was found to be most robust in the low-data regime (this could be trivially expanded to a shallow feedforward neural network).

(c) For the dimension of the latent space we adopt the choice $\dim(\boldsymbol{z}) = 0.5\cdot\dim(\boldsymbol{X})$. To motivate this choice, we note that the primary function of $\boldsymbol{z}$ is to induce an information bottleneck which is able to retain information about *effective* properties $\boldsymbol{X}$. A suitable choice however will always be problem-dependent (see also section 3.7).

[11] Excluding boundary conditions.
[12] Excluding boundary conditions.

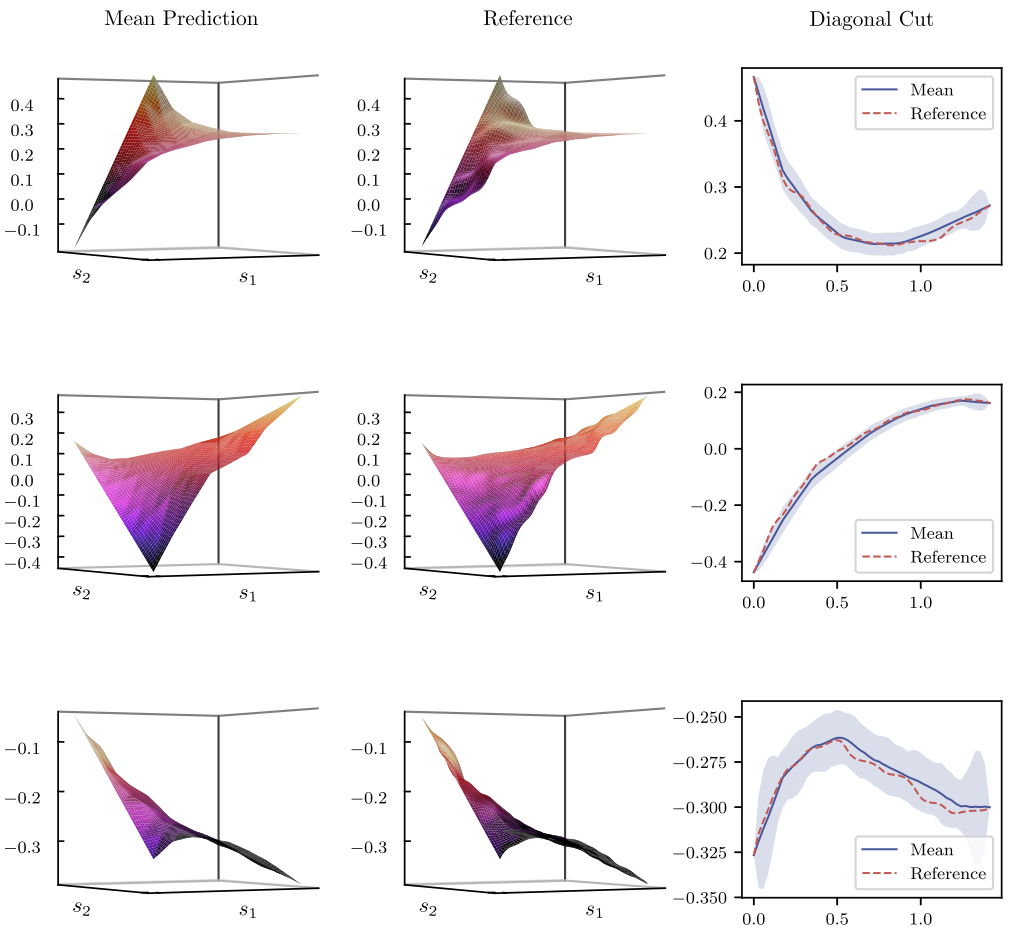


Fig. 5. The *left* column provides examples of the mean of the predictive posterior $p(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D})$ for various $\boldsymbol{x}$ not seen during training. The *middle* column contains the actual output $\boldsymbol{y}$ obtained by solving the FGM (ground truth / reference). Finally on the *right* column we compare the reference with the posterior predictive distribution by cutting along the diagonal of the unit square domain; the shaded area corresponds to the 95% credible interval ((64×64) FGM, (8×8) CGM, $l_\lambda = 0.04$).

The general implementation of the model leverages and intertwines both Fenics [81] as well as PyTorch [75]. The CGM and its adjoint have been fully embedded within the automatic differentiation framework of PyTorch, enabling the fast and parallel solution of the CGM on the GPU (i.e. in batches).

3.3. Virtual observables

Following the general discussion in section 2.2 on how domain knowledge can be introduced consistently in a probabilistic graphical model as artificial nodes (virtual observables), we discuss several types of such virtual observables $\mathcal{D}_{\mathcal{O}}$ derived from the governing equations. We are primarily interested in those that can inexpensively augment the training data and improve the predictive ability of the trained model even though they might provide *incomplete* or *partial* pieces of information at each input query point $\boldsymbol{x}^{(i_{\mathcal{O}})}$ about the underlying governing equations. This property (partial information) will be reflected in the fact that most constraints we consider only carry information about a small subset of dimensions in the $\boldsymbol{y}$-space. We note that when the virtual observables $\boldsymbol{o}(\boldsymbol{y}; \boldsymbol{x})$ are linear with respect to $\boldsymbol{y}$, then low-rank, closed-form updates for $\{q(\boldsymbol{y}^{(i_{\mathcal{O}})})\}_{i_{\mathcal{O}}=1}^{N_{\mathcal{O}}}$ (Equation (27)) can be employed. Detailed information on these technical matters is provided in Appendix B and in the appendices referenced in the ensuing discussion.

Weighted Residuals As discussed in the previous section, the method of weighted residuals can be used to enforce the governing equations. Hence we propose using Equation (50) as constraints that are probabilistically incorporated in the proposed model as discussed in section 2.2. We note that the use of weighted residuals of PDEs has also been advocated in deterministic machine-learning loss functions [46]. We consider two categories of residuals $r_w(\boldsymbol{y}; \boldsymbol{x})$ based on two different types of weight functions w. The latter can be thought of as the lens through which the governing equations are viewed.

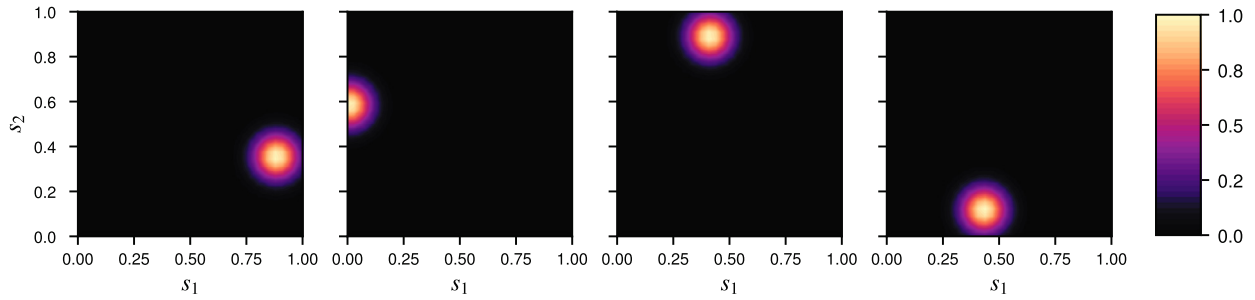


Fig. 6. Illustration of 4 randomly sampled radial basis-type weight functions (Eq. (55)) corresponding to the Randomized Residuals. Instead of using collocation points at which the PDE is enforced, we randomly sample Galerkin weight functions that enforce governing equations in a spatially-averaged sense.

The first type, which we call **Coarse-Grained Residuals**, employs weight functions w that correspond to the coarser discretization of the CGM. Due to the lower resolution of the corresponding mesh, they can be thought as enforcing the governing equations in a spatially-averaged sense. In particular and if we denote by $\boldsymbol{\Psi}(\boldsymbol{s}) = \{\Psi_m(\boldsymbol{s})\}_{m_1=1}^{M_1}$ the vector containing the shape-function of the CGM, we consider M_1 weight functions $\{w_{m_1}\}_{m_1=1}^{M_1}$ of the form[13]

$$w_{m_1}(\boldsymbol{s}) = \Psi_{m_1}(\boldsymbol{s}), \quad m_1 = 1, \dots, M_1 \,. \tag{54}$$

The second type of residuals considered and which we call **Randomized Residuals** are based on using M_2 radial basis-type functions as weight functions w, i.e.

$$w_{m_2}(\boldsymbol{s}) = \exp\left(-\frac{||\boldsymbol{s} - \boldsymbol{s}_{0,m_2}||^2}{\ell_{m_2}^2}\right), \qquad m_2 = 1, \dots, M_2 \,. \tag{55}$$

The scale parameters $\{\ell_{m_2}\}_{m_2=1}^{M_2}$ were set equal to 0.1 in subsequent investigations, and the centers $\{\boldsymbol{s}_{0,m_2}\}_{m_2=1}^{M_2}$ are sampled uniformly over the problem domain, i.e. $[0,1]^2$ (Fig. 6).

In contrast to the first type of residuals, these are capable of providing more localized information and over subdomains the size of which is determined by the scale parameters ℓ_{m_2} which can be adjusted accordingly. In the extreme where $\ell_{m_2} \to 0$, the weight function w_{m_2} becomes a Dirac-δ function and the corresponding constraint, a collocation-type one. The constraints associated with weighted residuals are enforced with infinite precision, i.e. $\sigma_c = 0$ in Equation (8).

Conservation (Flux) Constraint The second category of constraints that we employ can also be cast as a special case of weighted residuals, but operating instead directly on the conservation law (Equation (43)), i.e. on the flux variable $\boldsymbol{J}$ as in Equation (49). In particular, we make use of indicator functions of subdomains $\Omega_{m_3} \subseteq \Omega$ as weight functions w_{m_3}, i.e.

$$w_{m_3}(\boldsymbol{s}) = 1_{\Omega_{m_3}}(\boldsymbol{s}), \quad m_3 = 1, \dots, M_3 \,. \tag{56}$$

We note that in this case, Equation (49) reduces to

$$\int_{\partial\Omega_{m_3}} \boldsymbol{J}\, d\Gamma - \int_{\Omega_{m_3}} f\, d\boldsymbol{s} = 0 \,, \tag{57}$$

where the first integration is over the boundary of Ω_{m_3}. The subdomains Ω_{m_3} are selected to coincide with the finite elements of the CGM (Fig. 4). The flux $\boldsymbol{J}$ is computed using the constitutive law in Equation (44) from the discretized solution vector $\boldsymbol{y}$. Even though the spatial resolution of the weight functions is analogous to the ones in the Coarse-Grained Residuals above, the information the residuals of Equation (57) provide is of a different physical nature. Since not even the FGM satisfies such flux constraints perfectly, we learn the precision σ_c^{-2} (Equation (8)) with which these constraints are enforced by introducing a prior that promotes larger values (Appendix C). This is analogous to the well-known Automatic Relevance Determination (ARD, [70]) on the associated constraints.

Energy The final constraint that we make use of pertains to the type presented in Equation (10) (section 2.2) where the actual potential energy (Equation (53)) is employed. In contrast to the other constraints discussed, this provides *complete* information at each input query point, i.e. by minimizing V which implies fully enforcing the corresponding virtual observable, one can perfectly determine the solution vector $\boldsymbol{y}$. This precludes low-rank updates

[13] We always ensure these are *admissible*.

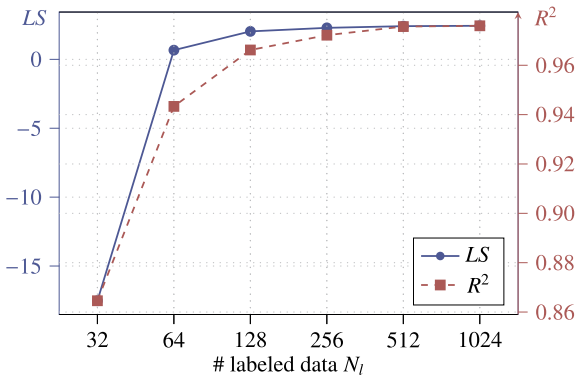


Fig. 7. Predictive performance in terms of the R^2 and LS metrics as a function of the number of labeled data points N_l ($N_u = N_{\mathcal{O}} = 0$), for $d_f = 32$ and $l_\lambda = 0.15$. Results have been averaged by repeatedly training the model on resampled data.

and makes the incorporation of this constraint more expensive. We provide details on how $\{q(\boldsymbol{y}^{(i_{\mathcal{O}})})\}_{i_{\mathcal{O}}=1}^{N_{\mathcal{O}}}$ is updated using stochastic second-order optimization in Appendix D.

3.4. Predictive performance and the effect of N_l

In the simplest scenario, the model is given access solely to a set of labeled data $\mathcal{D}_l = \{\boldsymbol{x}^{(i_l)}, \boldsymbol{y}^{(i_l)}\}_{i_l=1}^{N_l}$ (i.e. $N_u = N_{\mathcal{O}} = 0$). In the following we demonstrate as a baseline that the model generalizes well in the *small labeled data* regime, as a result of the information-bottleneck variables $\boldsymbol{z}$ as well as the CGM. We provide indicative samples of the mapping to the CGM inputs learned in Fig. 4 and indicative predictions for new inputs in Fig. 5.

As observed in Fig. 7, the model achieves very high scores with only $N_l = 128$ labeled data in terms of the R^2 (the largest possible value of R^2 is 1) and $N_l = 64$ in terms of the LS score. We observe that further increase of N_l results in minimal if not negligible improvement, i.e. the model has saturated. While alterations in the neural networks involved can be expected to change the particular values, we note that the saturation effect is a consequence of the limited capacity of the CGM which lies at the center of the model proposed. That is, even assuming an optimal choice for $\boldsymbol{\theta}$, the information bottleneck and the CGM implies that we can only predict the FGM output $\boldsymbol{y}$ up to a certain level of detail. Hence even if infinite (labeled) data were available, the predictive scores of the model would not improve further and the remaining pieces would be enveloped by the predictive uncertainty (see Fig. 5). On the other hand, if the CGM was removed and was substituted by a more expressive (and with more parameters) black-box model (e.g. another neural net), its predictive performance would not be as high with so few labeled data but would continue to increase (as much as its capacity would allow) with increasing N_l. This saturation effect arising from the CGM has also been observed in the discriminative model proposed in [58] where procedures for the adaptive refinement of the CGM were proposed. These were driven by the ELBO $\mathcal{F}$, which provides a natural score function for each model, but were not pursued in this work.

3.5. Effect of the amount and type of virtual observables

In the following, we demonstrate the benefits of the inclusion of virtual observables to the predictive performance of the proposed model. In order to quantify this benefit, we consider the posterior predictive density $p(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D}_l, \mathcal{D}_{\mathcal{O}})$ (section 2.6) as a function of labeled data $\mathcal{D}_l$ as well as of the virtual observables $\mathcal{D}_{\mathcal{O}} = \{\boldsymbol{x}^{(i)}, \hat{\boldsymbol{o}}^{(i)}\}_{i=1}^{N_{\mathcal{O}}}$. We omit in these experiments, *unlabeled data* $\mathcal{D}_u$ (i.e. $N_u = 0$), the effect of which will be examined in section 3.6. In particular, we examine the improvement in the predictive performance, i.e. in the metrics R^2 and LS (section 2.6.1), of the three baseline models (for $N_{\mathcal{O}} = 0$) corresponding to the following number of labeled data, i.e.

$$N_l = \{16,\ 32,\ 64\}\ , \tag{58}$$

when $N_{\mathcal{O}}$ virtual observables are added, where:

$$N_{\mathcal{O}} = \{32,\ 64,\ 128,\ 196,\ 256\}\ . \tag{59}$$

Furthermore, we examine the effect of the different types of virtual observables by considering the following three categories:

- **CGR**: At each input query point $\boldsymbol{x}^{(i_{\mathcal{O}})}$, $M_1 = 25$ Coarse-Grained Residuals (Equation (54)) are observed .
- **Hybrid**: At each input query point $\boldsymbol{x}^{(i_{\mathcal{O}})}$ the CGR ($M_1 = 25$), a set of randomized weighted residuals ($M_2 = 60$, Equation (55)) and the conservation of flux ($M_3 = 32$, Equation (56)) are observed.
- **Energy**: At each input query point $\boldsymbol{x}^{(i_{\mathcal{O}})}$ the potential energy is observed.

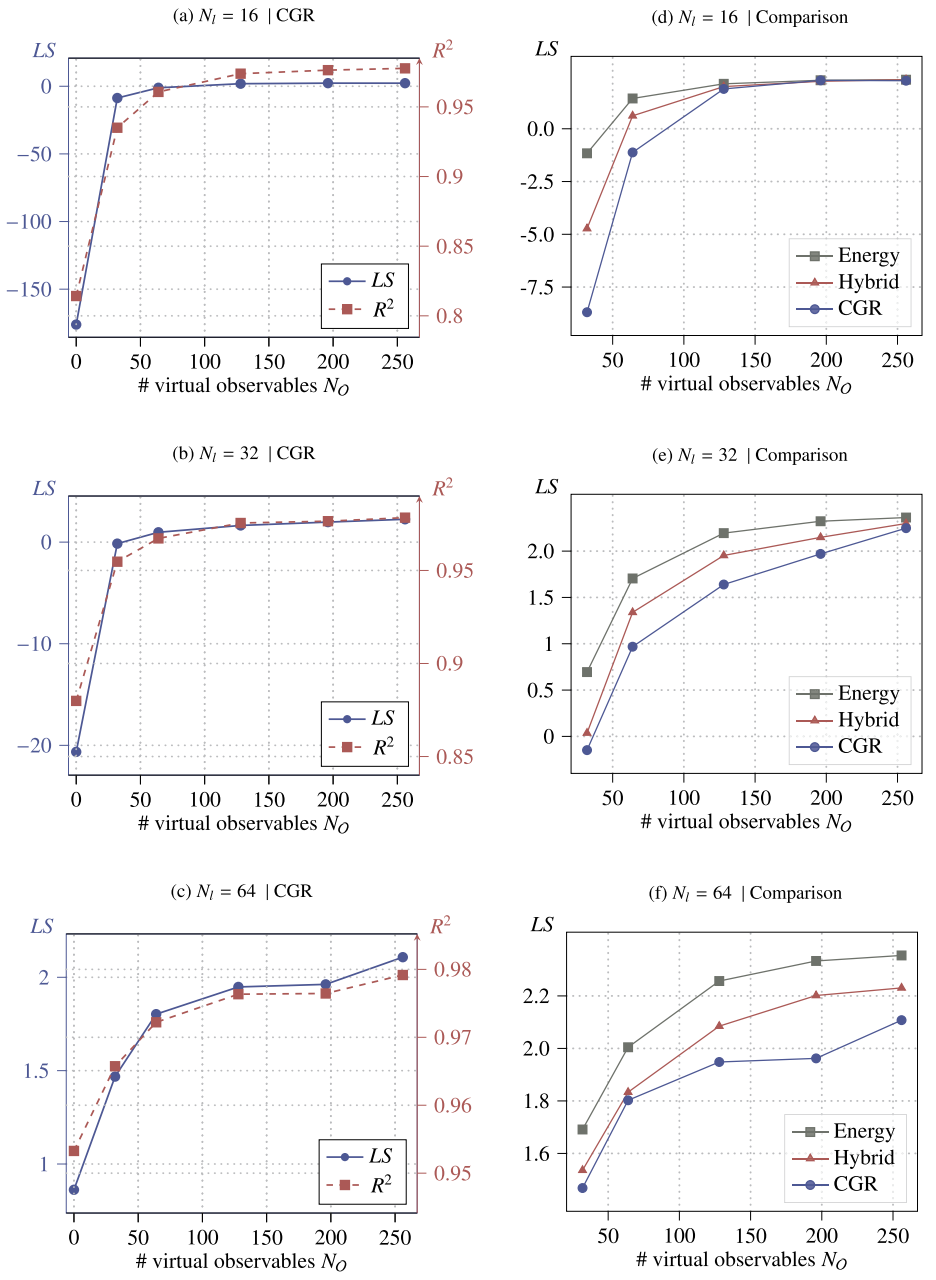


Fig. 8. Left Column: Predictive performance of a model trained on N_l labeled data, $N_{\mathcal{O}}$ virtual observables of type CGR ($N_u = 0$). Right Column: Comparison of predictive performance in terms of the *LS* metric with respect to 3 different *types* of virtual observables. The baseline performance for $N_{\mathcal{O}} = 0$ has been removed to improve clarity but the corresponding values can be found in the left column as well as Fig. 7. Results have been averaged by repeatedly training the model on resampled data.

We report results in Fig. 8, where the left column depicts the evolution of the R^2 and *LS* for different values of $N_{\mathcal{O}}$ and for virtual observables of the CGR type. One can readily observe that, for all three N_l values (i.e. number of labeled data), the introduction of the domain-knowledge in the form of these residual-type constraints leads to a significant improvement of the model's predictive accuracy. Furthermore, with the virtual observables introduced, one can attain with only $N_l = 16$ predictive performance scores that in Fig. 7 required $N_l = 512$ labeled data i.e. a significant reduction in the number of times the FGM needs to be solved. As one would perhaps expect, the gains from the virtual observables are more pronounced for small numbers of labeled data, i.e. when the model still struggles to generalize based on the too few labeled data points and therefore has more room to improve. Despite the fact that these virtual observations $\hat{\boldsymbol{o}} \in \mathbb{R}^{32}$ only provide partial information, the model is still able to leverage this to improve upon its predictive performance.

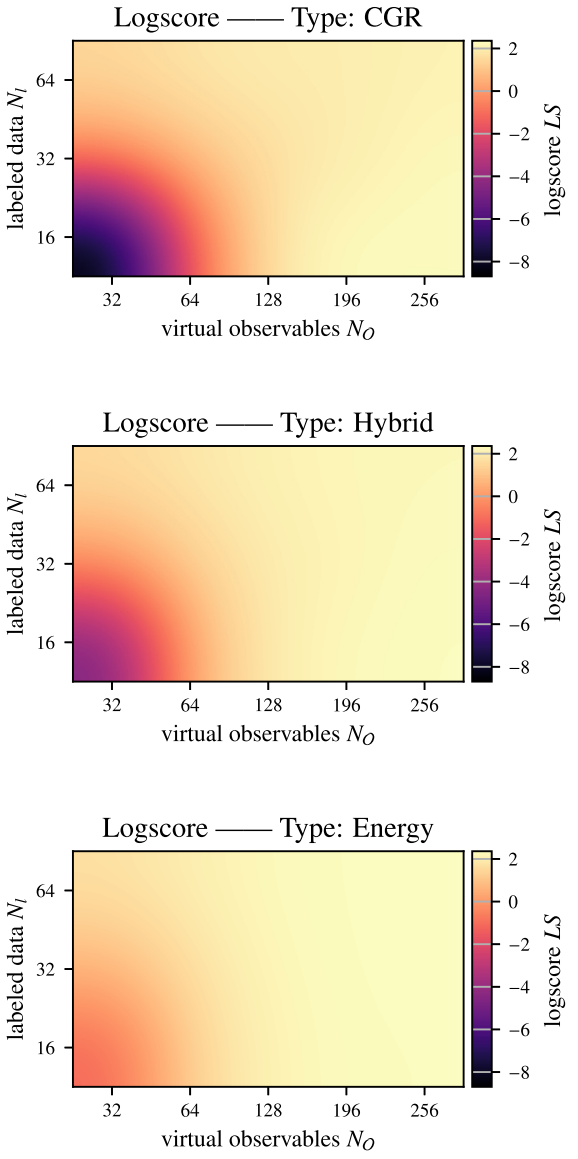


Fig. 9. LS score as function of N_l (number of labeled data) and $N_{\mathcal{O}}$ (number of virtual observables). Results have been averaged by repeatedly training the model on resampled data.

In the right column of Fig. 8 we expand upon these results by considering *different types* of virtual observables and by quantifying the impact of their informational content on the model's predictive performance. We note that the energy virtual observables have the most striking benefit which is expected as they provide complete information on the associated FGM output. Secondly, the *Hybrid*-type seems to yield a higher improvement in the model's predictive score as compared to the CGM-type. Finally in Fig. 9, we provide additional details by depicting the LS metric as a function of both $N_{\mathcal{O}}$ and N_l.

3.6. Effect of unlabeled data

In this section we study the effect of unlabeled data $\mathcal{D}_u = \{\boldsymbol{x}^{(i)}\}_{i=1}^{N_u}$, i.e. semi-supervised learning, in the model's predictive accuracy. To this end we investigate the predictive posterior $p(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D}_u, \mathcal{D}_l)$ as the number of unlabeled data N_u increases. We re-emphasize that unlabeled data are inexpensive to obtain (i.e. just inputs) and if the generative model proposed can exploit their informational content in improving its predictive ability, this would be of high utility.

In Fig. 10 we present the evolution of predictive metrics R^2 and LS as a function of the number of labeled data N_l for two models. The blue line corresponds to no unlabeled data, i.e. $N_u = 0$, whereas the red line corresponds to $N_u = 256$ unlabeled data. In both Figures the benefit of $\mathcal{D}_u$ can be clearly observed. The unlabeled data contribute in the identification of the lower-dimensional encoding $\boldsymbol{z}$, i.e. a compressed description of the input $\boldsymbol{x}$ which in turn informs the prediction of

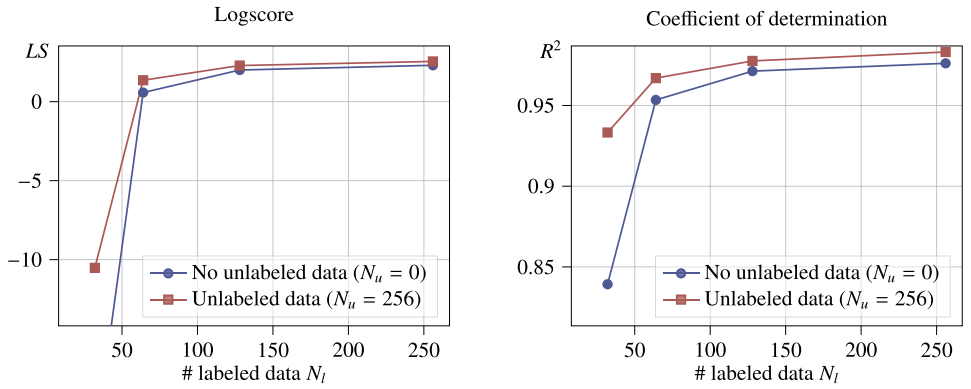


Fig. 10. A model trained on a certain number of labeled data N_l is compared to a model which in addition had access to $N_u = 256$ unlabeled data points, the latter achieving consistently better performance. Results have been averaged by repeatedly training the model on resampled data.

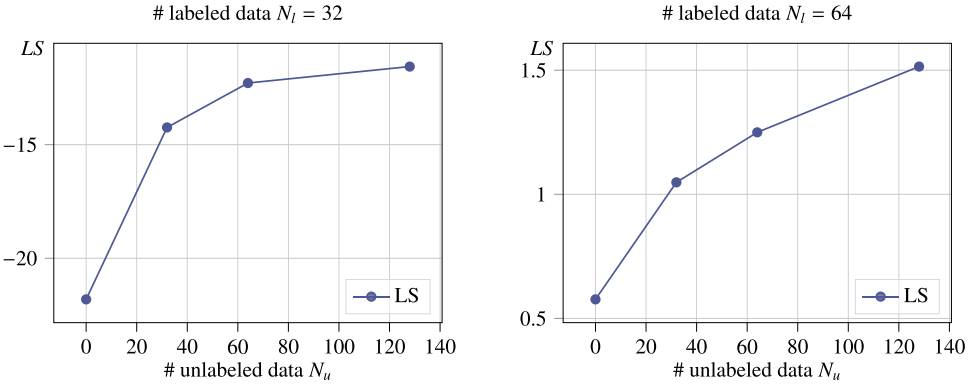


Fig. 11. The predictive performance of the generative model as a function of the number of unlabeled data N_u for $N_l = 32$ (left) and $N_l = 64$ (right). Results have been averaged by repeatedly training the model on resampled data.

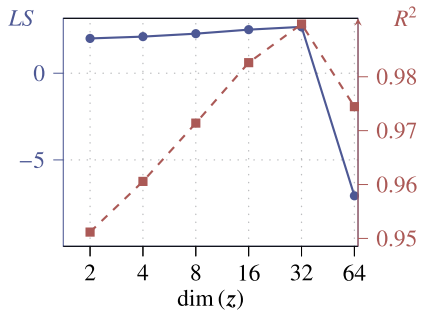
the output $\boldsymbol{y}$ through $\boldsymbol{X}$ i.e. the CGM (Fig. 2). As one can also observe, the benefit of unlabeled data decreases the higher N_l (i.e. the number of labeled data) is. This is not unexpected as the room for improvement is smaller for higher N_l.

Fig. 11 conveys similar information by varying the number of unlabeled data points while N_l if fixed (either to $N_l = 32$ or $N_l = 64$). The improvement in the predictive performance due to addition of unlabeled data points can be clearly observed. We further note that this improvement is always less than what one would attain with additional labeled data or with virtual observables (Fig. 9).

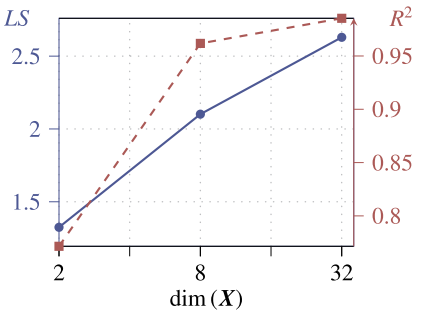
3.7. Effect of the lower-dimensional encoding and the CGM

In the following we provide a brief exposition of the effect of the dimension of the latent encoding $\boldsymbol{z}$ and the state variables $\boldsymbol{X}$ (and $\boldsymbol{Y}$) of the CGM on the predictive accuracy. In Fig. 12a we alter the dimension of the dim ($\boldsymbol{z}$) and clearly observe the existence of the information bottleneck, i.e. there exists threshold for $dim(\boldsymbol{z})$ up to which an improvement of the generative model is observed (for a fixed number of labeled data $N_l = 256$ and $N_u = 256$). After this threshold, the predictive capability of the model deteriorates, since the ability to retain more information in the latent encoding $\boldsymbol{z}$ is now superseded by the inability of the model to generalize well in the low-data-regime about the (increasingly complex) mappings linking the latent space to effective properties $\boldsymbol{X}$ and random field discretizations $\boldsymbol{x}$.

With regards to the dimension of $\boldsymbol{X}$ (or equivalently the resolution of the CGM), and as one would perhaps expect, there is an improvement in performance, as long as the dimension of the latent space as well as the number of data points afford the ability to exploit the increasing expressivity of the CGM. In Fig. 12b we illustrate the improvement of the predictive performance as the discretization of the CGM is increased from $dim(\boldsymbol{X}) = 2$ (i.e. a CGM resolution of (1×1) - $d_c = 1$) to $dim(\boldsymbol{X}) = 32$ (i.e. a CGM resolution of (4×4) - $d_c = 4$). The resolution of the FGM was (32×32) (i.e. $d_f = 32$) and the results presented were obtained for $N_l = 512, N_u = 512$ and $dim(\boldsymbol{z}) = 32$. We refer also to Fig. 4 for an illustration of the learned inputs $\boldsymbol{X}$ for various resolutions of the CGM.



(a) Predictive performance as a function of the dimension of the latent space dimension $Q = \dim(z)$; bottleneck occurs after $\dim(z) = 32$ (CGM = (4×4), $N_l = 256$, $N_u = 256$).



(b) Predictive performance as a function of $\dim(\boldsymbol{X})$, corresponding to the level of resolution of the computational domain by the CGM ($N_l = 512, N_u = 512, N_O = 0, dim(z) = 32$).

Fig. 12. Effect of the dimension of the latent encoding $\boldsymbol{z}$ and $\boldsymbol{X}$ on the predictive performance. Results have been averaged by repeatedly training the model on resampled data.

Table 1
(a) Different BCs considered, and (b) Predictive performance *LS* score obtained when training a model under the BCs indicated by the row and tested on the BCs indicated by the column.

Boundary Conditions

	A	B	C	D
a_0	0	1	$\mathcal{U}(-0.5, 0.5)$	0
a_1	0	1	0	Beta (2, 5)
a_2	1	0	0	−Beta (2, 5)
a_3	1	0	$\mathcal{U}(-0.5, 0.5)$	0

Logscore *LS*

prediction on / trained on	A	B	C	D
A	1.30	1.30	2.61	2.34
B	1.40	1.40	2.64	2.39
C	1.26	1.24	2.75	2.30
D	1.17	1.13	2.44	2.42

3.8. Effect of different BCs

In the following we evaluate the predictive performance of the model in an *extrapolative* setting, i.e. when the model is asked to provide predictions for boundary conditions not observed during training. To this end we consider the set of boundary conditions listed in Table 1a, where the coefficients a_i refer to the definition of a parametric Dirichlet B.C. as given in Equation (47) (for any a_i we specify either a fixed value, or a distribution of it to be randomly sampled from).

In Table 1b we report the *LS* score obtained on a validation dataset ($N_v = 256$). In all cases the model was trained on $N_l = 512$ labeled and $N_u = 2048$ unlabeled data (with $N_{\mathcal{O}} = 0$) using an amortized encoder. The diagonal terms correspond to predictive scores on the same BCs as the ones used for training (interpolative), whereas the off-diagonal ones to scores obtained on different BCs than the ones used for training (extrapolative). The results indicate that the predictive performance does not significantly depend upon the type of boundary condition the model has been trained on, i.e. the predictive performance in Table 1b only varies marginally across a column (BC used for training), and the variation is mostly determined (see row-wise), on which kind of boundary conditions we wish to make predictions.

3.9. Application: uncertainty propagation

As mentioned earlier, many-query applications represent one of the main incentives for learning probabilistic surrogates. We consider here the case of uncertainty propagation where the goal is to compute statistics of Quantities of Interest (QoIs) associated with the output $\boldsymbol{y}$ when the input $\boldsymbol{x}$ is random with a density, say $p(\boldsymbol{x})$. In the following, we compare the reference solution for the density of such a scalar QoI $v(\boldsymbol{y})$ obtained by direct Monte Carlo employing $N_{MC} = 8192$ FGM runs with the marginal distribution $\tilde{p}(v|\mathcal{D})$ over the QoI obtained from the posterior predictive as

$$\tilde{p}(v|\mathcal{D}) = \int\int \delta(v - v(\boldsymbol{y}))\, p(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D})\, p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}\, \mathrm{d}\boldsymbol{y}\,, \tag{60}$$

where $p(\boldsymbol{x})$ is the sampling density of the FGM inputs. We chose as $v(\boldsymbol{y})$ the value of the solution of the PDE at the middle of our computational domain, i.e. at $\boldsymbol{s} = (0.5, 0.5)$. The generative model was trained with $N_u = 8192, N_l = 32$ and $N_{\mathcal{O}} = 256$ and the results obtained are illustrated in Fig. 13. The approximation $\tilde{p}(v|\mathcal{D})$ obtained from the probabilistic surrogate matches closely with the Monte Carlo reference. If we had adopted a fully Bayesian approach, i.e. if $p(\boldsymbol{\theta}|\mathcal{D})$ was captured beyond a point estimate, additional uncertainty bounds on the probability density function $\tilde{p}(v|\mathcal{D})$ could be derived [82]. Note that the approximate marginal distribution $\tilde{p}(v|\mathcal{D})$ as seen in Fig. 13 has been obtained by leveraging

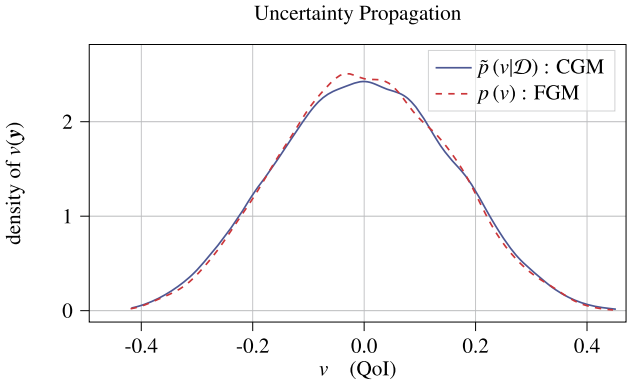


Fig. 13. The predictive posterior density $p(v|\mathcal{D})$ over the QoI $v(\boldsymbol{y})$ as compared with the Monte Carlo reference $p(v)$ obtained with $N_{MC} = 8192$ FGM solves. The model has been trained using $N_l = 32$ (compare this with N_{MC}), $N_u = 8192$ and $N_{\mathcal{O}} = 256$ *hybrid* virtual observables (see section 3.5). An amortized encoder was used for training and predictions.

the amortized encoder $p_\Phi(\boldsymbol{z}|\boldsymbol{x})$, such that each prediction merely requires to pass $\boldsymbol{x}$ through a neural network, followed by solving the CGM.

4. Conclusions

We have proposed a generative probabilistic model for constructing surrogates for PDEs characterized by high-dimensional parametric inputs $\boldsymbol{x}$ and high-dimensional outputs $\boldsymbol{y}$. In the following we summarize the most important and novel characteristics which enable the model to generalize in the small (labeled) data setting

- it learns the joint density $p(\boldsymbol{x}, \boldsymbol{y})$ in contrast to the conditional $p(\boldsymbol{y}|\boldsymbol{x})$ that most *discriminative* models in the literature target. As a result, it can make use of *unlabeled* data (i.e., only inputs $\boldsymbol{x}$) and enable training in a semi-supervised fashion.
- the choice of a latent variable model defines an information-bottleneck, and as such provides a mechanism to identify salient features of the random vector $\boldsymbol{x}$ which are predictive of the output. In other words, the information bottleneck forces the model to identify a small set of (complex and non-linear) features, which exhibit high mutual information with the solution $\boldsymbol{y}$. This is achieved by maximizing of the ELBO which yields an encoding $p_\theta(\boldsymbol{z}|\boldsymbol{x})$ in the latent space that is 'rich' in information concerning the output $\boldsymbol{y}$ we wish to predict [83].
- it employs a coarse-grained model at its core which serves to further tighten the information-bottleneck between the high-dimensional inputs $\boldsymbol{x}$ and outputs $\boldsymbol{y}$. We have demonstrated how such models can be flexibly constructed by coarsening the FGM and have shown that this can lead to superior predictive performance in the *small labeled data* regime as well as under extrapolative conditions (i.e., boundary conditions *not* used during training). Part of the complexity of the expensive FGM is absorbed by the CGM which in turn reduces the dependence on (labeled) data. Alternatively one may regard this as an additional constraint imposed upon the generative model, as the mean predictions for $p(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D})$ are restricted to the manifold that is defined by a coarse-grained physical process [47].
- it makes use of domain knowledge in the form of constraints/equalities or functionals that govern the original physical problem. These are incorporated in the likelihood in a fully Bayesian fashion as *virtual observables* and can lead to significant performance gains while reducing further the need for expensive, labeled data. Furthermore, we have demonstrated the beneficial effect of such virtual observables even in cases where they only provide incomplete/partial information of the FGM solution vector.
- it yields a predictive posterior density that can be used not only for point estimates, but for quantifying the predictive uncertainty as well. The latter is most often neglected in similar efforts but it is an unavoidable consequence of any coarse-graining or dimensionality-reduction or reduced-order-modeling scheme that is trained on finite amounts of data.

The proposed modeling framework provides a fertile ground for several extensions. Apart from the obvious refinement, both in terms of breadth and depth, of the neural networks employed, these improvements would involve:

- the automatic discovery of the dimension of the latent variables $\boldsymbol{z}$ as well as of the CGM. In the latter case, this could involve the dimension of the state variables $\boldsymbol{X}, \boldsymbol{Y}$ as well as the model-form itself, i.e. the relation between $\boldsymbol{X}$ and $\boldsymbol{Y}$. As previously mentioned, the ELBO $\mathcal{F}$ could serve as the driver for such investigations since it quantifies the plausibility of the data under a given model by balancing the quality of the fit with the model's complexity [84,58].
- active learning in terms of unlabeled data and virtual observables. As it has been demonstrated, such data provide valuable information in improving the model. It is not necessary though that all inputs $\boldsymbol{x}$ or pairs of inputs and virtual

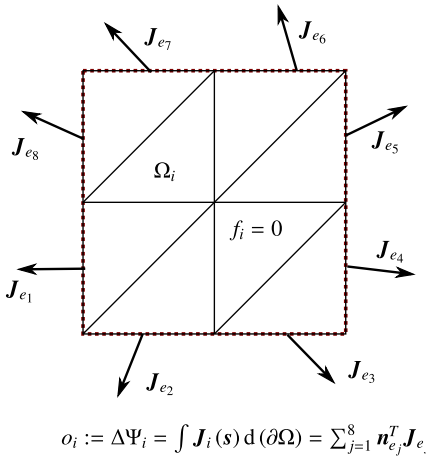


Fig. A.14. If the source term f_i associated with subdomain Ω_i is zero, then the integrated flux across the boundary should net to zero. The discrepancy of this flux $o_i := \Delta\Psi_{\Omega_i}$ corresponds to a virtual observable (equality constraint) introduced as artificial node in our probabilistic graphical model.

observables $(\boldsymbol{x}, \hat{\boldsymbol{o}})$ provide the same information. A critical component in improving the overall training efficiency would be to employ active learning schemes [85] in order to adaptively select the inputs and/or virtual observables (e.g. weight functions) at each step that are most informative. We note that such a scheme and in the context of a *deterministic* PDE-surrogate has been proposed in [46]. Extensions in the probabilistic setting advocated could also make use of the ELBO in selecting from a vocabulary of options, the one that would lead to the largest increase in $\mathcal{F}$.

CRediT authorship contribution statement

Maximilian Rixner: Conceptualization, Data curation, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft. **Phaedon-Stelios Koutsourelakis:** Conceptualization, Methodology, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Encoding conservation laws as equality constraints

A wide range of PDEs imply physical conservation laws, i.e. the governing equation state that some quantity Ψ is conserved and unchanging. Since this holds for any arbitrary subdomain $\Omega_i \subset \Omega$ and time interval we may express this in integral form [86] as

$$\Delta\Psi_{\Omega_i}(t) = \frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_i}\int \Psi(\boldsymbol{s},t)\mathrm{d}\Omega_i + \int_{\partial\Omega_i} \boldsymbol{J}_i(\boldsymbol{s},t)\mathrm{d}(\partial\Omega_i) - \int_{\Omega_i} f_i(\boldsymbol{s},t)\mathrm{d}\Omega_i \tag{A.1}$$

where $\boldsymbol{s}$, $\boldsymbol{J}_i$ and f_i denote the spatial coordinates, (boundary) flux and source term of subdomain Ω_i, respectively. We may introduce this physical conservation constraint into our model by introducing $o_i = \Delta\Psi_{\Omega_i}$ as a virtual observable. A virtual observable may then for instance correspond to violation of energy conservation resulting from the CGM predictions, entering into the probabilistic model by virtue of a zero-mean virtual Gaussian likelihood, i.e. $o_i := \Delta\Psi_{\Omega_i} \sim \mathcal{N}\big(0, \tau_i^{-1}\big)$. For our steady-state elliptic problem with no time-dependence Equation (A.1) simplifies to

$$\Delta\Psi_{\Omega_i} = \int_{\partial\Omega_i} \boldsymbol{J}_i(\boldsymbol{s})\mathrm{d}\Gamma - \int_{\Omega_i} f_i(\boldsymbol{s})\cdot\mathrm{d}\Omega_i\,, \tag{A.2}$$

which states that the net-flow across the boundary $\partial\Omega_i$ must be equal to production specified by the source term (see also Equation (43) and (57)). With $u(\boldsymbol{s}) = \sum_{j=1}^{d_y} \varphi_j^u(\boldsymbol{s})\, y_j$ given by a Finite Element discretization of local (linear) shape functions defined on some triangulation $\mathcal{T}$ of the computational domain, Equation (A.2) results in a linear constraint, since the flux $\boldsymbol{J}(\boldsymbol{s})$ reduces to an element-wise constant quantity (see Fig. A.14), enabling us to compute

$$\int_{\partial\Omega_i} \boldsymbol{J}(\boldsymbol{s})\mathrm{d}\Gamma = \sum_{j=1}^{N_e} \boldsymbol{n}_{e_j}^T \boldsymbol{J}_{e_j}\,, \tag{A.3}$$

where the element-wise constant flux $\boldsymbol{J}_{e_i} = \boldsymbol{B}^{(i)}\boldsymbol{y}$ is linear in $\boldsymbol{y}$ with $\boldsymbol{B}^{(i)} \in \mathbb{R}^{2\times d_y}$, and we sum over all finite elements comprising the subdomain Ω_i (assuming a compliant mesh). As such for the choice of M subdomains $\Omega_i, i = 1, ..., M$ we may define as virtual observable a vector $\boldsymbol{o}(\boldsymbol{y};\boldsymbol{x})$ (where the i-th entry corresponds to $\Delta\Psi_{\Omega_i}$) which can be expressed as

$$\boldsymbol{o}(\boldsymbol{y};\boldsymbol{x}) = \boldsymbol{\Gamma}(\boldsymbol{x})\,\boldsymbol{y} - \boldsymbol{\alpha}(\boldsymbol{x})\,, \tag{A.4}$$

with the entries of $\boldsymbol{\Gamma}(\boldsymbol{x})$ deriving from (A.3) and $\boldsymbol{J}_{e_i} = \boldsymbol{B}^{(i)}\boldsymbol{y}$, while $\alpha_i = \int_{\Omega_i} f_i(\boldsymbol{s})\cdot\mathrm{d}\Omega_i$.

Appendix B. Low-rank mean-field updates for virtual observables

While in principle the entire model can be trained using stochastic variational inference[14] as outlined in Algorithm 1, for linear equality constraints we are able to perform closed-form mean-field updates for $q(\mathcal{Y}_\mathcal{O})$, providing both additional insight as well as computationally efficient updates. For any ensemble of linear physical constraints enforced with a certain precision $\boldsymbol{\Lambda}$ we may write

$$\boldsymbol{o}(\boldsymbol{y},\boldsymbol{x}) := \boldsymbol{\Gamma}(\boldsymbol{x})\,\boldsymbol{y} - \boldsymbol{\alpha}(\boldsymbol{x}) \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{\Lambda}^{-1}\right) \qquad \boldsymbol{\Gamma}(\boldsymbol{x}) = \left[\boldsymbol{\gamma}_1(\boldsymbol{x})^T, ..., \boldsymbol{\gamma}_M(\boldsymbol{x})^T\right] \in \mathbb{R}^{M\times d_y} \tag{B.1}$$

where the entries of $\boldsymbol{\Gamma}(\boldsymbol{x})$ and $\boldsymbol{\alpha}(\boldsymbol{x})$ derive from the particular choice of constraint and the underlying physics at a query point $\boldsymbol{x}$ (see section 3.3). The precision matrix $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, ..., \lambda_M)$ is chosen diagonal, such that the set of parameters $\boldsymbol{\tau}$ governing the enforcement of our constraints follows as $\boldsymbol{\tau} = \{\lambda_i\}_{i=1}^M$. Given the assumed structure of the variational approximation $q_\xi(\boldsymbol{\theta}, \mathcal{R})$ (see Equation (27)), note that the optimal $q^*(\mathcal{Y}_\mathcal{O})$ follows by integrating out all other factors of q_ξ [70]

$$\begin{aligned}
\log q^*(\mathcal{Y}_\mathcal{O}) &= \mathbb{E}_{\tilde{q}_\xi}\left[\log\left(p\left(\hat{\mathcal{O}}\middle|\mathcal{Y}_\mathcal{O}, \mathcal{X}_\mathcal{O}, \boldsymbol{\Lambda}\right) p(\mathcal{Y}_\mathcal{O}|\mathfrak{X}_\mathcal{O}, \boldsymbol{\theta})\, p(\mathfrak{X}_\mathcal{O}|\mathcal{Z}_\mathcal{O}, \boldsymbol{\theta})\, p(\mathcal{X}_\mathcal{O}|\mathcal{Z}_\mathcal{O}, \boldsymbol{\theta})\, p(\mathcal{Z}_\mathcal{O})\, p(\boldsymbol{\theta})\right)\right] \\
&= \mathbb{E}_{\tilde{q}_\xi}\left[-\sum_{i_\mathcal{O}=1}^{N_\mathcal{O}}\left[\frac{1}{2}\left(\boldsymbol{y}^{(i_\mathcal{O})} - \boldsymbol{h}\left(\boldsymbol{X}^{(i_\mathcal{O})}\right)\right)^T \boldsymbol{S}_{\boldsymbol{y}}^{-1}\left(\boldsymbol{y}^{(i_\mathcal{O})} - \boldsymbol{h}\left(\boldsymbol{X}^{(i_\mathcal{O})}\right)\right)\right]\right] \\
&+ \mathbb{E}_{\tilde{q}_\xi}\left[-\sum_{i_\mathcal{O}=1}^{N_\mathcal{O}}\left[\frac{1}{2}\left(\boldsymbol{\Gamma}\left(\boldsymbol{x}^{(i_\mathcal{O})}\right)\boldsymbol{y} - \boldsymbol{\alpha}\left(\boldsymbol{x}^{(i_\mathcal{O})}\right)\right)^T \boldsymbol{\Lambda}\left(\boldsymbol{\Gamma}\left(\boldsymbol{x}^{(i_\mathcal{O})}\right)\boldsymbol{y} - \boldsymbol{\alpha}\left(\boldsymbol{x}^{(i_\mathcal{O})}\right)\right)\right]\right] + \text{const.}\,,
\end{aligned} \tag{B.2}$$

where $\hat{\mathcal{O}} = \{\hat{\boldsymbol{o}}\}_{i_\mathcal{O}=1}^{N_\mathcal{O}}$ comprises all virtual observations and $\tilde{q}_\xi$ denotes all other factors of the structured mean-field approximation aside from $q(\mathcal{Y}_\mathcal{O})$, i.e. $q_\xi = q(\mathcal{Y}_\mathcal{O})\tilde{q}_\xi$. Inspecting Equation (B.2) we find that it is linear-quadratic in $\boldsymbol{y}$, which implies a Gaussian $q(\boldsymbol{y}^{(i_\mathcal{O})}) = \mathcal{N}(\boldsymbol{\mu}^{(i_\mathcal{O})}, \boldsymbol{\Sigma}^{(i_\mathcal{O})})$ at every query point with mean and covariance implicitly defined by (for $i_\mathcal{O} = 1, ..., N_\mathcal{O}$)

$$\begin{aligned}
{\boldsymbol{\Sigma}^{(i_\mathcal{O})}}^{-1}\boldsymbol{\mu}^{(i_\mathcal{O})} &= \boldsymbol{\Gamma}\left(\boldsymbol{x}^{(i_\mathcal{O})}\right)^T \boldsymbol{\Lambda}\left(\boldsymbol{x}^{(i_\mathcal{O})}\right)\boldsymbol{\alpha}\left(\boldsymbol{x}^{(i_\mathcal{O})}\right) + \left\langle \boldsymbol{S}_{\boldsymbol{y}}^{-1}\right\rangle\left\langle \boldsymbol{h}\left(\boldsymbol{Y}\left(\boldsymbol{X}^{(i_\mathcal{O})}\right);\boldsymbol{\theta}\right)\right\rangle \\
{\boldsymbol{\Sigma}^{(i_\mathcal{O})}}^{-1} &= \boldsymbol{\Gamma}\left(\boldsymbol{x}^{(i_\mathcal{O})}\right)^T \boldsymbol{\Lambda}\boldsymbol{\Gamma}\left(\boldsymbol{x}^{(i_\mathcal{O})}\right) + \left\langle \boldsymbol{S}_{\boldsymbol{y}}^{-1}\right\rangle,
\end{aligned} \tag{B.3}$$

where $\langle\cdot\rangle$ denotes an expectation with respect to all remaining factors of the variational approximation $\tilde{q}_\xi$. Given our model choices (Eqs. (16) - (19)), the expectation of the precision matrix $\langle \boldsymbol{S}_{\boldsymbol{y}}^{-1}\rangle$ is constrained to be diagonal while the matrix $\boldsymbol{\Gamma}\left(\boldsymbol{x}^{(i)}\right)^T\boldsymbol{\Lambda}\boldsymbol{\Gamma}\left(\boldsymbol{x}^{(i)}\right)$ with $\boldsymbol{\Gamma} \in \mathbb{R}^{M\times d_y}$ exhibits low-rank structure. This low-rank structure reflects the fact that we only have introduced *partial* or *incomplete* information, and as such the constraints are only informative for a certain (low-dimensional) subspace. It simultaneously allows us to cheaply incorporate this physical knowledge into our model, since we may exploit the low-rank structure and use the Woodbury matrix identity to obtain mean vector and covariance matrix of the Gaussians $q\left(\boldsymbol{y}^{(i_\mathcal{O})}\right) = \mathcal{N}\left(\boldsymbol{\mu}^{(i_\mathcal{O})}, \boldsymbol{\Sigma}^{(i_\mathcal{O})}\right)$ at a cost $\mathcal{O}\left(M^3\right)$, i.e. numerical expense of updating $q\left(\boldsymbol{y}^{(i)}\right)$ depends on the number of enforced constraints rather than the dimension of $\boldsymbol{y}$. Making use of the Woodbury matrix identity one finds

$$\boldsymbol{\Sigma}^{(i_\mathcal{O})} = \left\langle \boldsymbol{S}_{\boldsymbol{y}}\right\rangle - \left\langle \boldsymbol{S}_{\boldsymbol{y}}\right\rangle\boldsymbol{\Gamma}\left(\boldsymbol{x}^{(i_\mathcal{O})}\right)^T {\boldsymbol{\Xi}^{(i_\mathcal{O})}}^{-1}\boldsymbol{\Gamma}\left(\boldsymbol{x}^{(i_\mathcal{O})}\right)\left\langle \boldsymbol{S}_{\boldsymbol{y}}\right\rangle, \tag{B.4}$$

[14] The required Jacobian of the virtual observables $\boldsymbol{o}(\boldsymbol{y},\boldsymbol{x})$ in order to propagate gradients simply reduces to the well-known Gateaux derivative, and is easily (as well as cheaply and parallelizable) obtained in most Finite Element frameworks (see e.g. *Unified Form Language* [87]).

where we have introduced the $M \times M$ matrix $\boldsymbol{\Xi}^{(i_{\mathcal{O}})} = \boldsymbol{\Gamma}(\boldsymbol{x}^{(i_{\mathcal{O}})})\langle \boldsymbol{S}_{\boldsymbol{y}} \rangle \boldsymbol{\Gamma}(\boldsymbol{x}^{(i_{\mathcal{O}})})^T + \boldsymbol{\Lambda}^{-1}$. In the limit case of components of the diagonal precision matrix $\boldsymbol{\Lambda}$ being infinite (i.e. absolute enforcement of the constraint), the result is an am improper Gaussian with rank-deficient covariance, i.e. the epistemic uncertainty of the epistemic uncertainty of the model collapses to a subspace which is completely in compliance with the enforced constraints; the update of $q(\mathcal{Y}_{\mathcal{O}})$ then becomes similar to the updates of Bayesian Conjugate Gradient (BCG) [88], which poses the solution of a linear equation system as a problem of probabilistic inference conditionally on the observance of a set of search directions.

Appendix C. Adaptively inferring finite precisions

For some physical constraints as, e.g., the flux constraint (Appendix A) it is neither plausible to assume infinite precision, nor do we a-priori know a suitable finite precision value with which to enforce the constraint. In such cases we may chose to treat the precision parameters $\tau = \{\lambda_m\}_{m=1}^{M}$ probabilistically as well. We propose to introduce a Gamma prior $\lambda_m \sim \Gamma(\alpha_0^{(m)}, \beta_0^{(m)})$ for each of the unknown precision values $\lambda^{(m)}$, or alternatively assume identical precision for all virtual observables (or subgroups thereof). For notational simplicity we discuss the latter case where all virtual observables are governed by a singular precision parameter λ

$$\lambda \sim \frac{\beta_0^{\alpha_0}}{\Gamma(\alpha_0)} \lambda^{\alpha_0 - 1} \exp(-\beta_0 \lambda) \,. \tag{C.1}$$

The variational approximation is extended to include $q(\lambda)$, and following the same approach as for the closed-form updates of $q(\mathcal{Y}_{\mathcal{O}})$ in Appendix B, the optimal variational approximation $q^*(\lambda)$ can be found to be a Gamma distribution $\Gamma(\alpha, \beta)$, with parameters α and β given by

$$\alpha = \left(\sum_{i_{\mathcal{O}}=1}^{N_{\mathcal{O}}} \frac{1}{2} M \right) + \alpha_0 \qquad \beta = \frac{1}{2} \sum_{i_{\mathcal{O}}=1}^{N_{\mathcal{O}}} \mathbb{E}_{q\left(\boldsymbol{y}^{(i_{\mathcal{O}})}\right)} \left[\left|\left| \boldsymbol{o}\left(\boldsymbol{y}^{(i_{\mathcal{O}})}; \boldsymbol{x}^{(i_{\mathcal{O}})}\right) \right|\right|_2^2 \right] + \beta_0 \,, \tag{C.2}$$

where M the number of constraints at each query point governed by λ. For a linear constraint (B.1) and a Gaussian $q(\boldsymbol{y}^{(i_{\mathcal{O}})}) = \mathcal{N}(\boldsymbol{\mu}^{(i_{\mathcal{O}})}, \boldsymbol{\Sigma}^{(i_{\mathcal{O}})})$ as given by Equation (B.3) the expectation involved in finding β becomes tractable; otherwise they can be cheaply estimated using Monte Carlo. For the Gamma prior we chose $\alpha_0 = \beta_0 = 10^{-6}$.

Appendix D. Stochastic second order optimization for the energy-based virtual observables

The introduction of the energy as a virtual observable at $N_{\mathcal{O}}$ query point differs from the other constraints we considered, since in contrast to $M << d_y$ equality constraints it *fully* summarizes all the information about the governing equations. Specifically, for a Finite Element discretization of the linear elliptic PDE given by $\boldsymbol{K}(\boldsymbol{x})\,\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$, the energy can be expressed in discretized form as

$$V\left(\boldsymbol{y}^{(i_{\mathcal{O}})}, \boldsymbol{x}^{(i_{\mathcal{O}})}\right) = \frac{1}{2}\, \boldsymbol{y}^{(i_{\mathcal{O}})\,T}\, \boldsymbol{K}\left(\boldsymbol{x}^{(i_{\mathcal{O}})}\right) \boldsymbol{y}^{(i_{\mathcal{O}})} - \boldsymbol{f}\left(\boldsymbol{x}^{(i_{\mathcal{O}})}\right)^T \boldsymbol{y}^{(i_{\mathcal{O}})} \,, \tag{D.1}$$

and we find that the minimization of the quadratic potential $V\left(\boldsymbol{y}^{(i_{\mathcal{O}})}, \boldsymbol{x}^{i_{\mathcal{O}}}\right)$ is the dual problem to solving the linear equation system associated with the solution of the discretized PDE itself. The introduction of the energy similarly implies that the ELBO becomes a quadratic potential in $\boldsymbol{\mu}^{(i_{\mathcal{O}})}$; i.e. plausibility of the model as scored by the ELBO now depends on the energy state obtained for predictions at all $N_{\mathcal{O}}$ query points. With the virtual likelihood defined by a Exponential distribution as given by Equation (12) and following the same mean-field approach as in Appendix B, the optimal $q\left(\boldsymbol{y}^{(i_{\mathcal{O}})}\right) = \mathcal{N}\left(\boldsymbol{\mu}^{(i_{\mathcal{O}})}, \boldsymbol{\Sigma}^{(i_{\mathcal{O}})}\right)$ is similarly found to be a Gaussian with mean and covariance defined by (for $i_{\mathcal{O}} = 1, \ldots, N_{\mathcal{O}}$)

$$\boldsymbol{\Sigma}^{(i_{\mathcal{O}})\,-1}\, \boldsymbol{\mu}^{(i_{\mathcal{O}})} = \tau\, \boldsymbol{f}^{(i_{\mathcal{O}})} + \left\langle \boldsymbol{S}_{\boldsymbol{y}}^{-1} \right\rangle \left\langle \boldsymbol{h}\left(\boldsymbol{Y}\left(\boldsymbol{X}^{(i_{\mathcal{O}})}\right); \boldsymbol{\theta}\right)\right\rangle \qquad \boldsymbol{\Sigma}^{(i_{\mathcal{O}})\,-1} = \left\langle \boldsymbol{S}_{\boldsymbol{y}}^{-1} \right\rangle + \tau\, \boldsymbol{K}\left(\boldsymbol{x}^{(i_{\mathcal{O}})}\right) \,, \tag{D.2}$$

where τ is a precision or tempering parameter which governs the weight given to the virtual observables - for the limit case of τ approaching infinity, the belief about $\boldsymbol{y}^{i_{\mathcal{O}}}$ will entirely depend on the energy state and becomes independent of the probabilistic surrogate. In contrast to the enforcement of $M << d_y$ equality constraint, the precision matrix $\boldsymbol{\Sigma}^{(i_{\mathcal{O}})\,-1}$ is sparse but exhibits full-rank structure, precluding the possibility to perform low-rank updates. As such the maximization of the evidence lower bound as a quadratic potential w.r.t. $\boldsymbol{\mu}^{(i_{\mathcal{O}})}$ on first glance appears to be the dual problem to solving the linear PDE itself if no amortization is applied. Note however that

- the maximization of the ELBO defines a simplified transfer problem since $\mathrm{cond}(\tau \boldsymbol{K}(\boldsymbol{x}^{(i_{\mathcal{O}})}) + \langle \boldsymbol{S}_{\boldsymbol{y}}^{-1} \rangle) \leq \mathrm{cond}(\boldsymbol{K}(\boldsymbol{x}^{(i_{\mathcal{O}})}))$, i.e. the probabilistic surrogate implicitly acts as a preconditioner. When optimizing the evidence lower bound we merely use the energy to *correct* the predictions of the surrogate and to pull them gradually in the right direction, instead of solving the PDE from scratch. This suggests an approach where one slowly tempers τ during training
- knowledge is transferred and mediated by the probabilistic model, as opposed to solving $N_{\mathcal{O}}$ entirely disjoint problems

- we are not intrinsically interested in $q(\boldsymbol{y})$ but only to the extend to which it is able to inform our probabilistic surrogate, (i.e. learn the parameters $\boldsymbol{\theta}$ of the generative model). As such, due to the inherent irreducible error introduced by the CGM, beyond a certain point there is no benefit in increasing τ (which, e.g., can be seen to correspond to the tolerance parameter of iterative solvers)

Despite this, it has to be noted that the incorporation of this inequality constraint is comparably much more expensive and bears more resemblance to the original forward problem defined by the FGM. Since we want to avoid solving the equation system implied by Equation (D.2) directly, we constrain the covariance matrix $\boldsymbol{\Sigma}^{(i_{\mathcal{O}})}$ of the variational approximation $q(\boldsymbol{y}^{(i_{\mathcal{O}})}) = \mathcal{N}(\boldsymbol{\mu}^{(i_{\mathcal{O}})}, \boldsymbol{\Sigma}^{(i_{\mathcal{O}})})$ to be diagonal and chose to optimize $\mathcal{F}$ iteratively with respects to the parameters of $q(\boldsymbol{y}^{((i_{\mathcal{O}})})$ using second order stochastic optimization. Here we use randomized Newton [89,90], which can be seen to iteratively update parameters such that the iterates are as close as possible in the L2 norm, while simultaneously forcing the error to be zero with respect to a randomly sampled subspace (see *sketching-viewpoint* of [89]).

References

[1] P.S. Koutsourelakis, N. Zabaras, M. Girolami, Special Issue: Big data and predictive computational modeling, J. Comput. Phys. 321 (2016) 1252–1254, https://doi.org/10.1016/j.jcp.2016.03.028, http://www.sciencedirect.com/science/article/pii/S0021999116001807.
[2] G. Marcus, E. Davis, Rebooting AI: Building Artificial Intelligence We Can Trust, Pantheon, 2019.
[3] R. Stewart, S. Ermon, Label-free supervision of neural networks with physics and domain knowledge, in: Thirty-First AAAI Conference on Artificial Intelligence, 2017.
[4] P.S. Koutsourelakis, Stochastic upscaling in solid mechanics: an exercise in machine learning, J. Comput. Phys. 226 (1) (2007) 301–325.
[5] R.G. Ghanem, P.D. Spanos, Stochastic Finite Elements: A Spectral Approach, Springer, New York, 1991, http://cds.cern.ch/record/1622736.
[6] D. Xiu, G. Karniadakis, The Wiener–Askey polynomial chaos for stochastic differential equations, SIAM J. Sci. Comput. 24 (2) (2002) 619–644, https://doi.org/10.1137/S1064827501387826.
[7] D. Xiu, J. Hesthaven, High-order collocation methods for differential equations with random inputs, SIAM J. Sci. Comput. 27 (3) (2005) 1118–1139, https://doi.org/10.1137/040615201.
[8] X. Ma, N. Zabaras, An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations, J. Comput. Phys. 228 (8) (2009) 3084–3113, https://doi.org/10.1016/j.jcp.2009.01.006, http://www.sciencedirect.com/science/article/pii/S002199910900028X.
[9] G. Lin, A. Tartakovsky, An efficient, high-order probabilistic collocation method on sparse grids for three-dimensional flow and solute transport in randomly heterogeneous porous media, Adv. Water Resour. 32 (5) (2009) 712–722, https://doi.org/10.1016/j.advwatres.2008.09.003, http://www.sciencedirect.com/science/article/pii/S0309170808001632; Special Issue: Dispersion in Porous Media.
[10] S. Torquato, B. Lu, Chord-length distribution function for two-phase random media, Phys. Rev. E 47 (1993) 2950–2953, https://doi.org/10.1103/PhysRevE.47.2950.
[11] J. Hesthaven, G. Rozza, B. Stamm, Certified Reduced Basis Methods for Parametrized Partial Differential Equations, Springer Briefs in Mathematics, Springer International Publishing, ISBN 978-3-319-22469-5, 2016, https://www.springer.com/de/book/9783319224695.
[12] A. Quarteroni, A. Manzoni, F. Negri, Reduced Basis Methods for Partial Differential Equations. An Introduction, La Matematica per il, vol. 3+2, Springer International Publishing, 2016, p. 92, http://infoscience.epfl.ch/record/218966.
[13] C.W. Rowley, T. Colonius, R.M. Murray, Model reduction for compressible flows using POD and Galerkin projection, Physica D 189 (1) (2004) 115–129, https://doi.org/10.1016/j.physd.2003.03.001, http://www.sciencedirect.com/science/article/pii/S0167278903003841.
[14] M. Guo, J. Hesthaven, Reduced order modeling for nonlinear structural analysis using gaussian process regression, Comput. Methods Appl. Mech. Eng. 341 (2018) 807–826, https://doi.org/10.1016/j.cma.2018.07.017, http://www.sciencedirect.com/science/article/pii/S0045782518303487.
[15] J. Hesthaven, S. Ubbiali, Non-intrusive reduced order modeling of nonlinear problems using neural networks, J. Comput. Phys. 363 (2018) 55–78, https://doi.org/10.1016/j.jcp.2018.02.037, http://www.sciencedirect.com/science/article/pii/S0021999118301190.
[16] J.N. Kani, A.H. Elsheikh, Dr-rnn: a deep residual recurrent neural network for model reduction, arXiv preprint, 2017, arXiv:1709.00939.
[17] Q. Wang, N. Ripamonti, J.S. Hesthaven, Recurrent neural network closure of parametric POD-Galerkin reduced-order models based on the Mori-Zwanzig formalism, J. Comput. Phys. (2020) 109402.
[18] K. Lee, K.T. Carlberg, Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders, J. Comput. Phys. 404 (2020) 108973.
[19] C. Rasmussen, C. Williams, Gaussian Processes for Machine Learning, Adaptive Computation and Machine Learning, MIT Press, Cambridge, MA, USA, 2006.
[20] I. Bilionis, N. Zabaras, B.A. Konomi, G. Lin, Multi-output separable Gaussian process: towards an efficient, fully Bayesian paradigm for uncertainty quantification, J. Comput. Phys. 241 (2013) 212–239, https://doi.org/10.1016/j.jcp.2013.01.011, http://www.sciencedirect.com/science/article/pii/S0021999113000417.
[21] I. Bilionis, N. Zabaras, Bayesian uncertainty propagation using Gaussian processes, in: Handbook of Uncertainty Quantification, Springer International Publishing, Cham, ISBN 978-3-319-12385-1, 2017.
[22] A. O'Hagan, M. Kennedy, Predicting the output from a complex computer code when fast approximations are available, Biometrika 87 (1) (2000) 1–13, https://doi.org/10.1093/biomet/87.1.1.
[23] P.S. Koutsourelakis, Accurate uncertainty quantification using inaccurate computational models, SIAM J. Sci. Comput. 31 (5) (2009) 3274–3300, https://doi.org/10.1137/080733565.
[24] M. Raissi, P. Perdikaris, G.E. Karniadakis, Inferring solutions of differential equations using noisy multi-fidelity data, J. Comput. Phys. 335 (2017) 736–746, https://doi.org/10.1016/j.jcp.2017.01.060, http://www.sciencedirect.com/science/article/pii/S0021999117300761.
[25] P. Perdikaris, D. Venturi, J.O. Royset, G.E. Karniadakis, Multi-fidelity modelling via recursive co-kriging and Gaussian–Markov random fields, Proc. R. Soc. A, Math. Phys. Eng. Sci. 471 (2179) (2015) 20150018, https://doi.org/10.1098/rspa.2015.0018, https://royalsocietypublishing.org/doi/abs/10.1098/rspa.2015.0018.
[26] J. Nitzler, J. Biehler, N. Fehn, P.S. Koutsourelakis, W.A. Wall, A generalized probabilistic learning approach for multi-fidelity uncertainty propagation in complex physical simulations, arXiv:2001.02892, 2020.
[27] X. Yang, G. Tartakovsky, A. Tartakovsky, Physics-informed kriging: a physics-informed Gaussian process regression method for data-model convergence, arxiv e-print, 2018, https://arxiv.org/pdf/1809.03461.pdf.
[28] S. Lee, F. Dietrich, G. Karniadakis, I. Kevrekidis, Linking Gaussian process regression with data-driven manifold embeddings for nonlinear data fusion, arxiv e-print, 2018, https://arxiv.org/pdf/1812.06467.pdf.
[29] R. Tipireddy, A. Tartakovsky, Physics-informed machine learning method for forecasting and uncertainty quantification of partially observed and unobserved states in power grids, arxiv e-print, 2018, https://arxiv.org/pdf/1806.10990.pdf.

[30] M. Guo, J.S. Hesthaven, Reduced order modeling for nonlinear structural analysis using gaussian process regression, Comput. Methods Appl. Mech. Eng. 341 (2018) 807–826.
[31] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444, https://doi.org/10.1038/nature14539, http://www.nature.com/nature/journal/v521/n7553/full/nature14539.html.
[32] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016, http://www.deeplearningbook.org.
[33] J. Han, A. Jentzen, W. E, Solving high-dimensional partial differential equations using deep learning, Proc. Natl. Acad. Sci. 115 (34) (2018) 8505–8510, https://doi.org/10.1073/pnas.1718942115, https://www.pnas.org/content/115/34/8505.
[34] Y. Zhu, N. Zabaras, Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification, J. Comput. Phys. 366 (2018) 415–447.
[35] S. Mo, Y. Zhu, N. Zabaras, X. Shi, J. Wu, Deep convolutional encoder-decoder networks for uncertainty quantification of dynamic multiphase flow in heterogeneous media, Water Resour. Res. 55 (1) (2018) 703–728, https://doi.org/10.1029/2018WR023528, https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018WR023528.
[36] J. Sirignano, K. Spiliopoulos, DGM: a deep learning algorithm for solving partial differential equations, J. Comput. Phys. 375 (2018) 1339–1364, https://doi.org/10.1016/j.jcp.2018.08.029, arXiv:1708.07469.
[37] W. E, B. Yu, The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems, Commun. Math. Stat. 6 (1) (2018) 1–12, https://doi.org/10.1007/s40304-018-0127-z.
[38] M. Raissi, P. Perdikaris, G. Karniadakis, Physics informed deep learning (part I): data-driven solutions of nonlinear partial differential equations, arxiv e-print, 2017, https://arxiv.org/pdf/1711.10561.pdf.
[39] M. Raissi, G.E. Karniadakis, Hidden physics models: machine learning of nonlinear partial differential equations, J. Comput. Phys. 357 (2018) 125–141, https://doi.org/10.1016/j.jcp.2017.11.039, http://www.sciencedirect.com/science/article/pii/S0021999117309014.
[40] M. Raissi, P. Perdikaris, G. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707, https://doi.org/10.1016/j.jcp.2018.10.045, http://www.sciencedirect.com/science/article/pii/S0021999118307125.
[41] Y. Yang, P. Perdikaris, Conditional deep surrogate models for stochastic, high-dimensional, and multi-fidelity systems, arxiv e-print, 2019, https://arxiv.org/pdf/1901.04878.pdf.
[42] I. Lagaris, A. Likas, D. Papageorgiou, Neural-network methods for boundary value problems with irregular boundaries, IEEE Trans. Neural Netw. 11 (5) (2000) 1041–1049.
[43] M.A. Nabian, H. Meidani, A deep neural network surrogate for high-dimensional random partial differential equations, arXiv preprint, 2018, arXiv:1806.02957.
[44] C. Beck, W. E, A. Jentzen, Learning approximation algorithms for high-dimensional fully nonlinear partial differential equations and second-order backward stochastic differential equations, J. Nonlinear Sci. 29 (1563–1619) (2019) 1563–1619, https://doi.org/10.1007/s00332-018-9525-3.
[45] S. Karumuri, R. Tripathy, I. Bilionis, J. Panchal, Simulator-free solution of high-dimensional stochastic elliptic partial differential equations using deep neural networks, J. Comput. Phys. 404 (2020) 109120.
[46] R. Khodayi-Mehr, M.M. Zavlanos, VarNet: variational neural networks for the solution of partial differential equations, https://arxiv.org/abs/1912.07443, 2019.
[47] F.d.A Belbute-Peres, T. Economon, Z. Kolter, Combining differentiable pde solvers and graph neural networks for fluid flow prediction, in: International Conference on Machine Learning, in: PMLR, vol. 119, 2020, pp. 2402–2411.
[48] Y. Zhu, N. Zabaras, P.S. Koutsourelakis, P. Perdikaris, Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data, J. Comput. Phys. 394 (2019) 56–81.
[49] M. Frank, D. Drikakis, V. Charissis, Machine-learning methods for computational science and engineering, Computation 8 (1) (2020) 15, https://doi.org/10.3390/computation8010015, https://www.mdpi.com/2079-3197/8/1/15.
[50] J. Willard, X. Jia, S. Xu, M. Steinbach, V. Kumar, Integrating physics-based modeling with machine learning: a survey, arXiv:2003.04919, 2020.
[51] M. Mattheakis, P. Protopapas, D. Sondak, M. Di Giovanni, E. Kaxiras, Physical symmetries embedded in neural networks, arXiv:physics/1904089, 2020.
[52] J. Magiera, D. Ray, J.S. Hesthaven, C. Rohde, Constraint-aware neural networks for Riemann problems, J. Comput. Phys. 409 (2020) 109345.
[53] S. Brunton, J. Proctor, N. Kutz, Sparse identification of nonlinear dynamics (SINDy), in: APS Division of Fluid Dynamics Meeting Abstracts, 2016.
[54] Z. Long, Y. Lu, X. Ma, B. Dong, PDE-net: learning PDEs from data, arXiv preprint, 2017, arXiv:1710.09668.
[55] L. Felsberger, P. Koutsourelakis, Physics-constrained, data-driven discovery of coarse-grained dynamics, Commun. Comput. Phys. 25 (5) (2019) 1259–1301, https://doi.org/10.4208/cicp.OA-2018-0174.
[56] S. Kaltenbach, P.S. Koutsourelakis, Incorporating physical constraints in a deep probabilistic machine learning framework for coarse-graining dynamical systems, J. Comput. Phys. 419 (2020) 109673, https://doi.org/10.1016/j.jcp.2020.109673, http://www.sciencedirect.com/science/article/pii/S0021999120304472.
[57] C. Grigo, P.S. Koutsourelakis, Bayesian model and dimension reduction for uncertainty propagation: applications in random media, SIAM/ASA J. Uncertain. Quantificat. 7 (1) (2019) 292–323, https://doi.org/10.1137/17M1155867, https://epubs.siam.org/doi/abs/10.1137/17M1155867.
[58] C. Grigo, P.S. Koutsourelakis, A physics-aware, probabilistic machine learning framework for coarse-graining high-dimensional systems in the Small Data regime, J. Comput. Phys. 397 (2019) 108842, https://doi.org/10.1016/j.jcp.2019.05.053, http://www.sciencedirect.com/science/article/pii/S0021999119305261.
[59] O. Chapelle, B. Schölkopf, A. Zien, Semi-supervised learning, IEEE Trans. Neural Netw. 20 (3) (2009), https://doi.org/10.1109/TNN.2009.2015974.
[60] D.P. Kingma, S. Mohamed, D.J. Rezende, M. Welling, Semi-supervised learning with deep generative models, in: Advances in Neural Information Processing Systems, 2014, pp. 3581–3589.
[61] S. Yu, K. Yu, V. Tresp, H.P. Kriegel, M. Wu, Supervised probabilistic principal component analysis, in: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2006, pp. 464–473.
[62] M. Raissi, P. Perdikaris, G.E. Karniadakis, Machine learning of linear differential equations using gaussian processes, J. Comput. Phys. 348 (2017) 683–693.
[63] S. Levine, Reinforcement learning and control as probabilistic inference: tutorial and review, arXiv preprint, 2018, arXiv:1805.00909.
[64] M. Ortiz, L. Stainier, The variational formulation of viscoplastic constitutive updates, Comput. Methods Appl. Mech. Eng. 171 (3) (1999) 419–444, https://doi.org/10.1016/S0045-7825(98)00219-9, http://www.sciencedirect.com/science/article/pii/S0045782598002199.
[65] Q. Yang, L. Stainier, M. Ortiz, A variational formulation of the coupled thermo-mechanical boundary-value problem for general dissipative solids, J. Mech. Phys. Solids 54 (2) (2006) 401–424, https://doi.org/10.1016/j.jmps.2005.08.010, http://www.sciencedirect.com/science/article/pii/S0022509605001511.
[66] Y. Khoo, J. Lu, L. Ying, Solving parametric pde problems with artificial neural networks, arXiv preprint, 2017, arXiv:1707.03351.
[67] J. Paisley, D. Blei, M.I. Jordan, Variational Bayesian inference with stochastic search, in: J. Langford, J. Pineau (Eds.), 29th International Conference on Machine Learning, ICML, Edinburgh, UK, 2012.
[68] M.D. Hoffman, D.M. Blei, C. Wang, J. Paisley, Stochastic variational inference, J. Mach. Learn. Res. 14 (1) (2013) 1303–1347, http://dl.acm.org/citation.cfm?id=2502581.2502622.

[69] D.M. Blei, A. Kucukelbir, J.D. McAuliffe, Variational inference: a review for statisticians, J. Am. Stat. Assoc. 112 (518) (2017) 859–877.
[70] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
[71] D.P. Kingma, M. Welling, Auto-encoding variational Bayes, arXiv preprint, 2013, arXiv:1312.6114.
[72] H. Robbins, S. Monro, A stochastic approximation method, Ann. Math. Stat. (1951) 400–407.
[73] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, arXiv preprint, 2014, arXiv:1412.6980.
[74] U. Naumann, The Art of Differentiating Computer Programs: An Introduction to Algorithmic Differentiation, vol. 24, SIAM, 2012.
[75] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic Differentiation in Pytorch, 2017.
[76] D. Zhang, A coefficient of determination for generalized linear models, Am. Stat. 71 (4) (2017) 310–316.
[77] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
[78] Y. LeCun, P. Haffner, L. Bottou, Y. Bengio, Object recognition with gradient-based learning, in: Shape, Contour and Grouping in Computer Vision, Springer, 1999, pp. 319–345.
[79] B. Finlayson (Ed.), The Method of Weighted Residuals and Variational Principles, with Application in Fluid Mechanics, Heat and Mass Transfer, vol. 87, Academic Press, New York, ISBN 978-0-12-257050-6, 1972.
[80] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4700–4708.
[81] A. Logg, K.A. Mardal, G. Wells, Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book, vol. 84, Springer Science & Business Media, 2012.
[82] M. Schöberl, N. Zabaras, P.S. Koutsourelakis, Predictive collective variable discovery with deep bayesian models, J. Chem. Phys. 150 (2) (2019) 024109.
[83] N. Tishby, F.C. Pereira, W. Bialek, The information bottleneck method, arXiv preprint, arXiv:physics/0004057, 2000.
[84] C. Rasmussen, Z. Ghahramani, Occam's razor, in: Neural Information Processing Systems, vol. 13, 2001, pp. 294–300.
[85] K. Kandasamy, J. Schneider, B. Póczos, Query efficient posterior estimation in scientific experiments via Bayesian active learning, Artif. Intell. 243 (C) (2017) 45–56, https://doi.org/10.1016/j.artint.2016.11.002.
[86] K. Lee, K. Carlberg, Deep conservation: a latent dynamics model for exact satisfaction of physical conservation laws, arXiv preprint, 2019, arXiv:1909.09754.
[87] M.S. Alnæs, A. Logg, K.B. Ølgaard, M.E. Rognes, G.N. Wells, Unified form language: a domain-specific language for weak formulations of partial differential equations, ACM Trans. Math. Softw. 40 (2) (2014) 1–37.
[88] J. Cockayne, C. Oates, I. Ipsen, M. Girolami, A Bayesian Conjugate Gradient Method, 2018.
[89] R.M. Gower, P. Richtárik, Randomized iterative methods for linear systems, SIAM J. Matrix Anal. Appl. 36 (4) (2015) 1660–1690.
[90] R.M. Gower, D. Kovalev, F. Lieder, P. Richtárik, RSN: randomized subspace Newton, arXiv preprint, 2019, arXiv:1905.10874.