Technische Universität München
School of Engineering and Design

Physics-Informed and Data-Driven Probabilistic Modeling of Materials Systems

Maximilian Rixner, M.Sc.

Vollständiger Abdruck der von der School of Engineering and Design der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitz:

Prof. Dr.ir. Daniel J. Rixen

Prüfer*innen der Dissertation:

1. Prof. Ph.D. Phaedon-Stelios Koutsourelakis
2. Prof. Dr.-Ing. Wolfgang A. Wall

Die Dissertation wurde am 30.08.2022 bei der Technischen Universität München eingereicht und durch die School of Engineering and Design am 05.09.2023 angenommen.

Abstract

Random media and the process-structure-property chain generally define complex, high-dimensional and stochastic materials systems, posing a challenging setting for any prediction or optimization task. In this thesis, we pursue a Bayesian approach for learning and predicting the behavior of such systems, leveraging probabilistic machine learning methods to identify their effective, coarse-grained properties. A particular focus will be on limiting and mitigating the dependence on labeled data due to the computational cost of their procurement (through established numerical discretization techniques). We achieve the prediction of high-dimensional stochastic systems defined by random media in the small-data domain by exploiting concepts such as physics-informed learning, active learning and semi-supervised learning. In addition to the data-parsimonious prediction of effective physical properties and behavior of random media, we also demonstrate the full stochastic inversion of the entire process-structure-property chain in a high-dimensional setting, thereby enabling the identification of optimal process parameters for computational materials design problems.

Zusammenfassung

Random Media und die Process-Structure-Property Kette definieren im Allgemeinen komplexe und hochdimensionale stochastische Materialsysteme, welche eine nicht-triviale Herausforderung für Prädiktion und Optimierung darstellen. In dieser Arbeit verfolgen wir einen Bayesschen Ansatz um das das Verhalten derartiger Systeme vorherzusagen, wobei effektive physikalische Eigenschaften durch den Einsatz von Methoden des wahrscheinlichkeitsbasierten Maschinellen Lernens identifiziert werden. Ein besonderer Fokus liegt hierbei in der Reduktion der Abhängigkeit von Daten (generiert durch konventionelle numerische Diskretisierungsansätze), aufgrund der damit assoziierten numerischen Kosten. Wir erreichen die Vorhersage hochdimensionaler stochastischer Systeme im Kontext von Random Media in der Small-Data Domain dank der Verwendung von Techniken wie Active Learning, Physics-Informed Learning und Semi-Supervised Learning. Zusätzlich zu der datensparsamen Vorhersage des physikalischen Verhaltens und Eigenschaften von Random Media demonstrieren wir die volle stochasische Inversion der gesamten Process-Structure-Property Kette in einer hochdimensionalen Anwendung, was die Identifizierung optimaler Prozessparameter für computergestütztes Design und Entwicklung neuer Materialien ermöglicht.

Contents

Abstract iii

Zusammenfassung iii

Contents iv

List of Figures vi

List of Tables vi

Acronyms vii

1 Introduction 1
- 1.1 Models, Probability, and Uncertainty . . . 1
- 1.2 Random Media and the Process-Structure-Property Chain . . . 3
- 1.3 Model Compression and Probabilistic Machine Learning . . . 6
- 1.4 Outline and Contributions . . . 7

2 Fundamentals 9
- 2.1 Probabilistic Models and Learning . . . 11
  - 2.1.1 Representation of Probabilistic Models . . . 13
  - 2.1.2 Example : Linear Hidden Markov Model . . . 13
- 2.2 Information Theory and Statistical Manifolds . . . 16
  - 2.2.1 Statistical Manifolds . . . 17
  - 2.2.2 Information Theory . . . 18
- 2.3 Bayes' Theorem . . . 21
  - 2.3.1 Updating States of Belief . . . 21
  - 2.3.2 Posterior Predictive and the Bayes Estimator . . . 24
  - 2.3.3 Bayes Factor . . . 24
  - 2.3.4 Uninformative and Improper Priors . . . 25
  - 2.3.5 Occam's razor and Geometric Complexity . . . 26
- 2.4 Approximate Probabilistic Inference . . . 31
  - 2.4.1 Variational Inference . . . 31

2.4.1.1 Evidence Lower Bound . . . 33
2.4.1.2 Mean-Field Approximation . . . 34
2.4.1.3 Expectation Maximization . . . 34
2.4.2 Sampling-based Inference . . . 36
2.5 Probabilistic Machine Learning and Deep Architectures . . . 38
2.5.1 Deep Learning as Approximate Variational Inference . . . 39
2.5.2 Example: Generative Latent Variable Model . . . 40
2.6 Stochastic Processes and Physical Systems . . . 43
2.6.1 Gaussian Random Fields . . . 44
2.6.2 Governing Equations . . . 45
2.7 Physics-Informed Learning and Differentiable Physics . . . 47
2.7.1 Automatic Differentiation and Backpropagation . . . 48
2.7.2 Differentiable Physics and the Adjoint Method . . . 49

3 Summary of Publications **53**
3.1 Paper A . . . 54
3.2 Paper B . . . 55

4 Discussion and Outlook **57**

Bibliography **63**

A Micro-to-Macro transition **91**

B Paper A **95**

C Paper B **125**

List of Figures

1.1 Binary two-phase random field realizations 3
1.2 Conceptual overview : process-structure-property chain 5

2.1 Hidden Markov Model 14
2.2 Occams' razor 28
2.3 Information-geometric illustration of Occam's razor 29
2.4 Information-projection onto a submanifold 32
2.5 Support and inductive bias of probabilistic models 39
2.6 Sequence of computations as a directed acyclic graph 48

A.1 Micro-to-macro transition 92

List of Tables

2.1 Interpretation of Bayes' factor 25

Acronyms

AIC Akaike Information Criterion

BE Balance Equation

BIC Bayesian Information Criterion

CE Constitutive Equation

CGM Coarse-Grained Model

CNN Convolutional Neural Network

DAG Directed Acyclic Graph

ELBO Evidence Lower Bound

EM Expectation-Maximization

ESS Effective Sample Size

FGM Fine-Grained Model

GAN Generative Adversarial Network

HMC Hamiltonian Monte Carlo

KLD Kullback-Leibler Divergence

LSTM Long Short-Term Memory

MAP Maximum A Posteriori

MALA Metropolis Adjusted Langevin Algorithm

MCMC Markov Chain Monte Carlo

MDL Minimum Description Length

MLE Maximum Likelihood Estimate

NN Neural Network

NUTS No-U-Turn Sampler

PDE Partial Differential Equation

PDF Probability Density Function

PGM Probabilistic Graphical Model

PMF Probability Mass Function

PSP Process-Structure-Property

P-S Process-Structure

S-P Structure-Property

RVE Representative Volume Element

SDF Spectral Density Function

SGD Stochastic Gradient Descent

SMC Squential Monte Carlo

SIS Sequential Importance Sampling

SPDE Stochastic Partial Differential Equation

SVI Stochastic Variational Inference

TLM Tangent Linear Model

VAE Variational Autoencoder

VB Variational Bayes

VI Variational Inference

1 Introduction

1.1 Models, Probability, and Uncertainty

At the core of engineering lies the pursuit to understand and predict complex physical systems or processes, and to subsequently leverage this insight to inform optimal design decisions or to exert control over the physical process (according to some specific, prescribed notion of optimality). The attainment of this goal is predicated on the construction and identification of *models*, which can be seen to formulate a hypothesis about the latent processes and governing principles underlying our observations. A model introduces a set of mechanistic assumptions about interdependencies of various entities or state variables and thereby serves to reduce the real-world complexity to a sufficiently useful representation. In such a setting, the utility of a model is judged according to its ability to provide a sufficiently accurate representation of the governing dynamics with respect to *specific* predictive tasks, i.e., its ability to explain observed data with sufficient accuracy. Consequently, the task of identifying, calibrating, or reasoning in the context of models lies at the heart of scientific discovery and engineering design. In its most general - and arguably also most elegant - form, models express interdependencies and relationships between various entities probabilistically [1, 2], i.e., posit non-deterministic relationships and articulate interdependencies in terms of plausibilities and degrees of belief. But even under the often simplified and more restrictive assumption of deterministic models, a probabilistic approach is still generally required in order to resolve both *aleatoric* and *epistemic* uncertainty, identify unknown model parameters or inputs (*inverse problem*), account for uncertain inputs to the model (*uncertainty propagation*), or tackle any other task in the context of limited information and/or stochasticity of inputs and model parameters [3, 4]. The uncertainty arising in such settings can in principle be differentiated with respect to their underlying cause [3]:

Aleatoric Uncertainty We refer to aleatoric uncertainty as stochasticity that cannot be attributed to a lack of knowledge or information, i.e., it is inseparably intertwined with the problem under consideration. While the delineation of aleatoric uncertainty can be an almost arbitrarily nuanced question, from a pragmatic perspective and for the purpose of this thesis we associate with aleatoric uncertainty the inherent

stochasticity of systems that cannot feasibly be reduced through the acquisition of additional information or data.

Epistemic Uncertainty Systems that exhibit deterministic behavior in principle will still be subject to epistemic uncertainty in a setting of finite data or limited information being available. As such, epistemic uncertainty may refer to uncertainty arising due to the limitation of finite information, finite data, or finite computational resources. The cost-effective or data-effective reduction of epistemic uncertainty pertains to the domains of active learning [5, 6] and Bayesian experimental design [7]. One notable example of epistemic uncertainty arises in the context of model compression, i.e., any surrogate, coarse-grained and reduced-order description of a physical process or model. The field of *probabilistic numerics* even goes so far to reconsider established, deterministic discretization techniques as probabilistic inference, providing probabilistic estimates and confidence intervals corresponding to the epistemic uncertainty arising from numerical discretization error [8, 9, 10, 11, 12].

Model Compression Even in cases where the physical process can in principle be resolved accurately in its entirety, considerations of the numerical cost may nonetheless lead to the requirement of reduced model descriptions or limited model evaluations. This constraint is particularly prevalent in the context of *many-query* applications, where repeated evolutions of the forward model are required. Prime examples are, e.g., given by calibration of complex multi-physics models (e.g. in biomechanics [13, 14, 15], or turbulent flow [16]). Despite decades of advancements in both computational resources as well as numerical methods, resolution of all relevant tempo-spatial scales (in particular for multi-query, multi-physics and multiscale applications) still quickly approaches the limitations of computational feasibility. Substituting a *fine-grained model* (FGM) with a *coarse-grained model* (CGM) - or any other suitable surrogate - hence defines a special case of epistemic uncertainty, induced by computational limitations.

The infeasibility of computationally fully resolving relevant physical processes by means of established numerical discretization techniques for many-query applications gives rise to the objective of this thesis to identify probabilistic surrogate models able to predict the behavior of such systems at a reduced cost, while still retaining sufficient predictive accuracy. Instead of an expert-defined model, a machine learning approach is employed to obtain a parsimonious and cheap representation of the physical process, leveraging data and/or a priori physical knowledge as a source of information for model identification (i.e., we posit an a priori hypothesis space and infer the candidate most suitable to explain observed data and/or physical constraints). Within such probabilistic models, uncertainties are then accounted for and propagated consistently irrespective of their respective origin or underlying cause. In this context we can also

note that according to Shannon [17], information can be regarded as the resolution of uncertainty, and hence the goal of learning and resolving uncertainties can be seen as dual in nature (or according to Jayes [1], *'probability distributions ... [are] carriers of incomplete information'*). In consequence - for a Bayesian setting - the process of *learning* or *generalizing* about a physical process by means of a model can be regarded as probabilistic inference conditional on our observations (indeed this notion is so general, that the free energy hypothesis [18, 19] seeks to express human cognition and action planning as complex hierarchical probabilistic inference processes). The particular objective under consideration in this thesis will be the identification of suitable probabilistic models for the prediction and control of stochastic materials systems in a high-dimensional and physics-constrained setting, which remains an area of ongoing research despite decades of advancements. The stochastic systems at which we aim our methodological efforts and discussion are defined by random media and the process-structure-property chain.

1.2 Random Media and the Process-Structure-Property Chain

We consider as a prototypical example a class of stochastic systems arising in the context of continuum thermodynamics [20], where stochastic variability is introduced by random spatial fluctuations of pertinent physical material properties (*random media*). Even with the governing equations of continuum thermodynamics assumed a priori already known, the physical system (characterized in our case by a partial differential operator) will become stochastic due to the inherent microscopic variability of the material. The spatial random variability of the material (as well as the thermodynamic state variables) can be modeled as random fields [21] and are characterized by generally complex spatial or even spatio-temporal correlation structures. Examples of random media are given for instance by binary two-phase and multiphase materials [22], for which two or more phases with distinct physical properties exhibit random spatial variability over the domain (see Figure 1.1). The formation of these phases

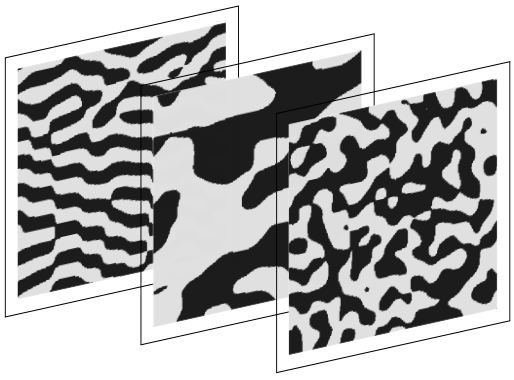


Figure 1.1: Three random field realizations for binary two-phase random heterogeneous media. Formally, we can describe this material by means of an indicator function $\mathbb{I}(s;\omega)$ depending on the spatial coordinate $s \in \mathbb{R}^d$ and the elementary event ω in conjunction with a suitable probability space $(\Omega, \mathcal{F}, \mathbb{P})$.

occurs itself according to complex physical processes, for binary two-phase material for instance it could be defined by the limiting case of the Cahn-Hilliard equation $\partial c/\partial t = D\nabla^2\left(c^3 - c - \lambda\nabla^2 c\right)$ converging towards phase separation (i.e. $c \in \{-1, 1\}$). Another notable example is given by polycrystalline materials [23], characterized by random variability of the grains (with the microstructure being defined by the sizes and orientations of the atomic lattice of the grains). Random media comprised of several distinct phases is referred to as *random heterogeneous media* [24]. While it is in principle possible to consider the *fine-grained* description of random media in full detail, resolving phenomena across all pertinent physical scales in a high-dimensional many-query setting will quickly deteriorate to a computationally infeasible setting. As such the endeavor in this thesis will be to identify parsimonious *features* of the fine-grained structure of the material on a microscopic level which are predictive of the *effective properties* [25] on a macroscopic scale (*structure-property-linkage*), or more generally, to identify a reduced description which still permits useful, probabilistic predictions about the system response under macroscopic loading (e.g., deformation or thermal response of the material). Finding a surrogate or reduced system that only depends on a comparably small set of features of the fine-scale material description which still enables to adequately capture the system response is referred to as *coarse-graining* or *model order reduction* [26, 27, 28], and implies the additional introduction of epistemic uncertainty in an already stochastic system due to model compression. Once a suitable surrogate is identified, it can be employed in any *many-query* setting to significantly reduce the computational burden. Additional complexity is introduced if we also consider how the statistical formation of microstructures is affected by *processing conditions*, with the stochastic linkage between processing conditions and material microstructures being referred to as the process-structure linkage. In conjunction with the previously discussed linkage between microstructures and their macroscopic properties and response, jointly this gives rise to the *process-structure-property* chain (see Figure 1.2). I.e., instead of relating individual microstructures to macroscopic properties, we seek to relate aggregate macroscopic properties to processing parameters governing the stochastic process underlying the formation of material microstructures. The inversion of this stochastic process-structure-property chain then corresponds to identifying the optimal process conditions that - as a statistical aggregate - give rise to physical properties that satisfy a certain criterion of optimality [29]. As this involves the *stochastic* inversion of both the process-structure as well as structure-property linkage (each corresponding to generally non-deterministic complex physical models), this poses a grand challenge and largely unsolved problem in a high-dimensional setting [30]. Both when predicting material properties and their response to physical loading in isolation, or in the more complex setting of the complete process-structure-property chain, the cost of repeated evaluations of the forward model will force the identification of suitable surrogates or coarse-grained descriptions of the physical process.

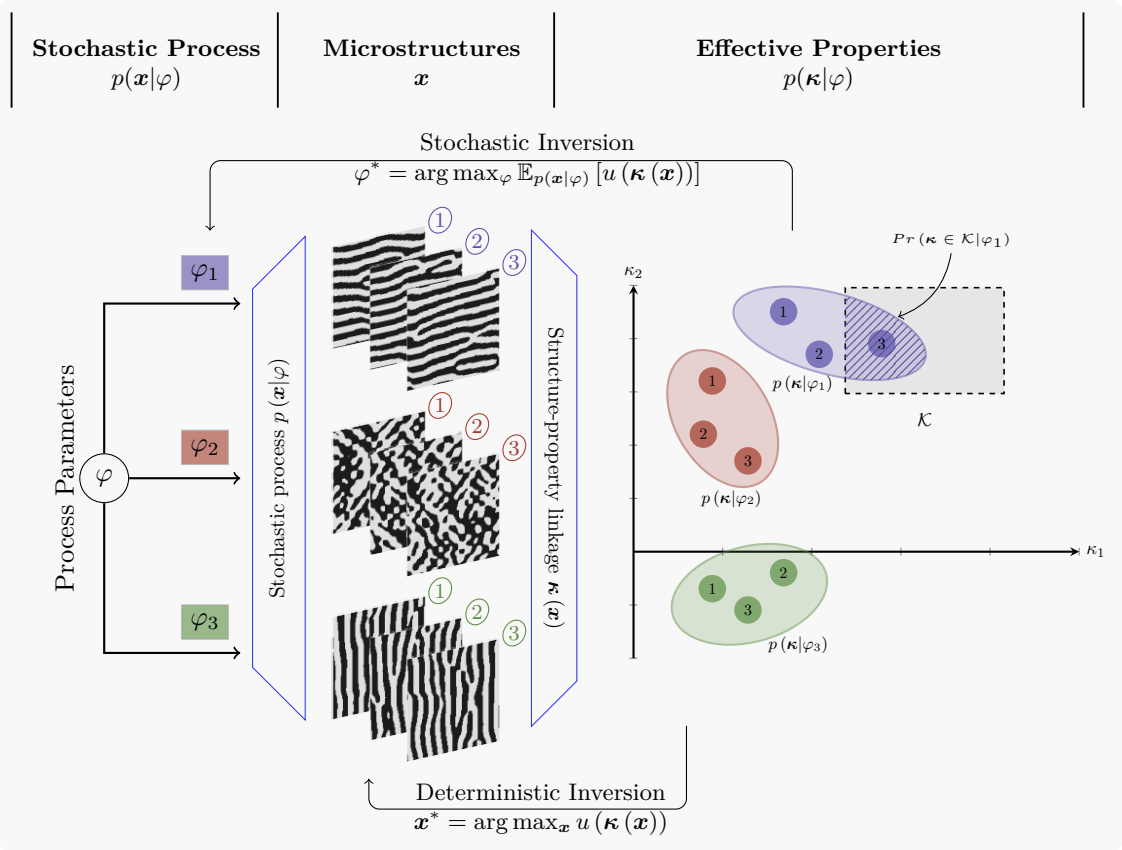


Figure 1.2: Conceptual overview of the process-structure-property chain and its stochastic inversion - reproduction of Figure 1 from [31], corresponding to section 3.2 of this thesis. Random heterogeneous media (here) in the form of binary two-phase microstructures arise as a stochastic process defined by process parameters. Learning the physical behavior or response of an individual microstructure can be considered as learning a structure-property mapping defined by a physical process, with microstructures themselves generated by a physical stochastic process (*process-structure linkage*). We aim to *(i)* learn the structure-property map in the small-data domain by means of a physics-informed, semi-supervised machine learning approach bottlenecked and physics-biased by a coarse-grained model and *(ii)* also consider full stochastic inversion of the entire process-structure-property chain by means of a conventional discriminative convolutional neural network surrogate embedded within an adaptive stochastic optimization framework (i.e., we seek to identify processing parameters which lead to optimal material properties of the microstructures, with the optimality criterion entailing an expectation over the generating stochastic process of the microstructures).

1.3 Model Compression and Probabilistic Machine Learning

Model compression refers to the endeavor of finding a surrogate or reduced description of a physical process, which nonetheless retains sufficient accuracy for predictive purposes [32]. The model under consideration for which we wish to identify a simplified description may exemplarily be given by a time-dynamical process, or in a stationary setting as a physical process which - upon discretization - can be regarded as relating inputs to outputs, e.g., a discretized differential operator $\mathcal{L}_h : \mathbb{R}^{d_x} \to \mathbb{R}^{d_y}$ (consider mapping discretized microstructures to their effective physical properties). Finding a suitable surrogate for such a physics-defined input-output system becomes particularly challenging in high-dimensional settings, and more so still when the problem under consideration is not easily amenable to dimensionality reduction (i.e., the probability mass is not concentrated on a low-dimensional manifold, or the manifold is highly nonlinear and complex). A wealth of surrogate-based approaches have been developed for such systems, but despite ongoing efforts, prediction and control of high-dimensional stochastic systems remain challenging and an area of ongoing research. One approach is to model the system response directly as a function of the inputs, as is exemplarily the case for Gaussian Process regression [33, 34] and (generalized) Polynomial Chaos [35, 36, 37]. While developments in this area such as sparse grid and stochastic collocation methods [38, 39, 40] will succeed in delaying the onset of the curse of dimensionality somewhat, for a sufficiently high-dimensional setting this approach still ceases to be feasible. In contrast, the vast field of projection-based methods [41, 42, 43, 44, 45, 46, 47, 48, 49, 50] constitute a model reduction approach that seeks to identify a reduced basis system to describe the physical process (in a probabilistic setting this would translate to the implicit assumption that most of the probability mass is concentrated on a lower-dimensional manifold, which can also be exploited in the inference process, e.g., [51, 52, 53]). As a prominent example for projection-based methods, Proper Orthogonal Decomposition (POD) [54] makes the assumption that the evolution of the state variables of a PDE can be expressed with respect to a linear subspace that is identified from a finite number of (sufficiently representative) snapshots. Despite recent attempts to combine this reduced order modeling approach with deep learning techniques [55, 56, 57, 58], in a high-dimensional setting the identification of a suitable reduced basis becomes highly non-trivial necessitating increasing amounts of (expensive) data. Another attempt to bypass the curse of dimensionality is represented by multi-level [59] and multi-fidelity [60, 61, 62, 63, 64] approaches, which introduce a hierarchy of solvers of varying numerical cost and either seek to exploit correlation or fuse information from different solvers to obtain predictions at a reduced cost. Their inherent constraint is however that they *presuppose* the existence of suitable a priori defined models of varying fidelity, and give no insight regarding their construction or identification. From the more recently prevailing perspective of *machine learning* [65] and *deep learning* [66, 67] (as well as the general emergence of data-driven approaches

in science and engineering [68, 69, 70]), the identification of a surrogate model for the previously discussed case of $\mathcal{L}_h : \mathbb{R}^{d_x} \to \mathbb{R}^{d_y}$ defines a *supervised learning* problem. The goal of machine learning in the context of model compression can then be regarded as the discovery of a much more *parsimonious* description of the physical process, ideally enabling considerably accelerated probabilistic reasoning and inference in the context of systems governed by physical laws. Learning (reduced) models for physical systems can be regarded as the accumulation of information regarding their behavior. While this information would most commonly spring from labeled data, given more recent advances it may also be partially derived from unlabeled data (semi-supervised learning), or is extracted directly from the governing equations and becomes infused into the probabilistic model as pseudo-observed artificial nodes (*virtual observables* [71, 28, 72]). When adopting a machine learning perspective it is important to realize that this field has generally evolved in the *big-data* domain, while in the context of physical models the acquisition of data is typically arduous and expensive (necessitating experimentation or expensive reference simulations with sufficiently accurate spatio-temporal discretizations). This in turn dictates that the development of probabilistic machine learning methods in such settings should be driven by the requirement of reducing dependence on (labeled) data as far as possible, i.e., providing the ability to operate in the *small-data* domain. The previously mentioned approach of extracting information from the governing equations directly into the model becomes of interest in such a setting, as it enables mitigating or altogether bypassing the dependence on expensive labeled data (*physics-informed machine learning* [73]). Similarly, careful choice of inductive model bias, as well as other previously mentioned techniques such as careful selection of data points (active learning) and incorporation of unlabeled data (semi-supervised learning), can reduce the dependence on expensive labeled training data.

1.4 Outline and Contributions

This thesis follows a probabilistic approach [1] and aims to develop novel numerical methods for the prediction and control of stochastic systems in a high-dimensional and data-scarce setting. Our effort will more specifically be aimed at predicting the physical properties of microstructures and random media, as well as the larger enveloping goal of achieving stochastic inversion of the entire process-structure-property chain using this predictive ability. The methodological approaches suggested and investigated can be interpreted as instances of probabilistic machine learning, seeking to exploit the combination of various concepts and ideas to alleviate the curse of dimensionality as well as the dependence on labeled data as far as possible (e.g., by adopting a principled approach to encode physical constraints in a probabilistic model as a priori knowledge, or by inducing inductive physical bias utilizing an embedded coarse-grained model). As this is a cumulative thesis with the methodological approach rooted in probabilistic

modeling and reasoning, the general introductory discussion in the following chapter 2 aims to introduce the most pertinent concepts and ideas underlying the papers in sections 3.1 and 3.2. Please note that we will forego a rigorous discussion of more mundane aspects of probability theory, i.e., assume basic familiarity and refer to the suitable literature on probability theory [1, 74, 2, 75, 76, 77, 78] and probabilistic machine learning [79, 65, 80]. With the basic foundation laid out, in section 3.1 we will introduce a generative model with inductive bias provided by an embedded coarse-grained model coupled to a latent variable representation of the random media. In addition to the inductive bias and the information bottleneck defined by the latent variables, predictive features are also informed by unlabeled data (semi-supervised learning) as well as by the injection of a priori knowledge of the governing equations encoded as *virtual observables* (i.e., as virtual nodes in our probabilistic graphical model). In section 3.2 we make use of a more conventional deterministic surrogate with convolutional architecture for the structure-property linkage in order to attain the inversion of the stochastic process-property-structure chain for a materials design problem in a high-dimensional setting. This ambitious goal [30] is achieved by the introduction of an inner-loop outer-loop approach with the incremental and adaptive refinement of the dataset informed by a problem-specific acquisition function. This active learning approach is coupled with a custom Variational-Bayes Expectation-Maximization algorithm that enables driving learning in this high-dimensional and intractable setting by providing low-noise Monte Carlo gradient estimates of the objective function.

2

Fundamentals

Doubt is not a pleasant condition,
but certainty is absurd.

Voltaire

The need for probabilistic reasoning not only arises in settings where stochastic phenomena are under investigation but in any setting where one wishes to reason in a context of finite and incomplete information. Probability theory therefore can be regarded as a rigorous theoretical framework of *'extended logic'* to conduct *'plausible reasoning'* [1] beyond a mere deductive setting. The necessity for such a framework arises due to the limitation of deterministic reasoning - consider exemplarily the following Boolean statement (or *syllogism*)

$$A \rightarrow B \tag{2.1}$$

which Jaynes [1] characterizes as a *logical* consequence which does not permit to draw inference upon A, even upon having observed B to be true. A *weak syllogism* is introduced by the notion that knowing B may not permit an absolute statement about the veracity of A, but that A should become more *plausible* after having observed B. This gives rise to the fundamental question, of whether one may consistently and quantitatively ascertain the plausibility of the statement A in such a setting. In a seminal paper [81] published in 1946, Cox showed that introducing a set of axioms for this framework of 'plausible' reasoning (the axioms here are reproduced as formulated by MacKay [75])

Axiom 1 *'Degrees of belief can be ordered [...] and in consequence can be mapped onto real numbers'*

Axiom 2 *'The degree of belief in a proposition x and its negation $\overline{x}$ are related. There is a function f such that $B(x) = f[B(\overline{x})]$'*

Axiom 3 *'The degree of belief in a conjunction of propositions x, y (x AND y) is related to the degree of belief in the conditional proposition $x|y$ and the degree of belief in the proposition y. There is a function g such that $B(x, y) = g[B(x|y), B(y)]$'*

yields a mathematical framework isomorphic to probability theory as following from Kolmogorov's axioms [82], and that moreover, it constitutes the only mathematical framework in compliance with the consistency requirements expressed by these axioms. A more informal and colloquial formulation of these desiderata for a framework of plausible reasoning as stated by Arnborg and Sjödin [83]:

Divisibility and comparability: *'The plausibility of a statement is a real number and is dependent on information we have related to the statement'*

Common sense: *'Plausibilities should vary sensibly with the assessment of plausibilities in the model'*

Consistency *'If the plausibility of a statement can be derived in two ways, the two results must be equal'*

When probability theory is used in the context of plausible reasoning, it is commonly referred to as the *Bayesian* perspective [83]. In the Bayesian interpretation of probability, *'probability is a measure of the degree of belief about a proposition'*, or equivalently *'a state of knowledge in presence of partial information'* [84]. This importantly implies that we cannot only argue about probabilities as relative frequencies of repeatable events but that instead, we can *consistently* reason about *any* given hypothesis in a non-deterministic or finite-information setting (enabling quantitative assessments as degrees of belief or plausibility). In such a setting the inability to make absolute statements generally either derives from *epistemic* uncertainty (lack of knowledge), or *aleatory* uncertainty (inherent randomness), i.e., *'probability explains the limitations of our knowledge of truth'* [85]. From this must necessarily follow that probabilistic rea-

soning aims to obtain a *'measure or characterization of truth'* [85], conditional on the observations and finite information available to us (in any setting where truth defines a meaningful concept - consider, e.g., the recovery of governing equations from limited data of observed physical phenomena [86, 87, 88]). The demarcation line between epistemic and aleatoric uncertainty impeding this pursuit of truth can be a subject of contention (consider e.g. turbulent flow), but within the Bayesian framework, uncertainty is treated and resolved independently of its root cause. Furthermore, it is to be emphasized that the theoretical framework of probabilistic reasoning derived from Cox's axioms does not merely provide a set of expeditious tools useful for the solution of specific problems. Instead, it gives rise to what Jaynes referred to as the *'logic of science'* [1], i.e., a *principled* way to reason probabilistically about the world. In the context of the more confined setting of this thesis, i.e. considering stochastic system governed by physical laws, probabilistic reasoning allows - among other things - to draw inference and learn from data (*machine learning*) [65, 80], quantity limits of knowledge and assess behavior of stochastic systems (*uncertainty quantification*) [89, 4, 90], assess reliability [91] and consistently deal with epistemic uncertainty [3, 92], formulate inverse problems [93, 94], perform data analysis [95], enable decision-making [74, 96], as well as stochastic optimization [97] and control [98]. While of course there are assumptions underlying the Bayesian approach of probabilistic reasoning (e.g., the axioms, model choices and prior distributions), we follow MacKay in his observation that *'you cannot do inference without making assumptions'* [75], which in the Bayesian probabilistic framework are simply made transparent. We will continue this section by providing a brief exposition about probabilistic models as well as the representation of their structure, and conclude by discussing as an illustrative example a Hidden Markov Model at the end of this section (before then moving on to discuss the nature of information encoded in such models in section 2.2.)

2.1 Probabilistic Models and Learning

A probabilistic model [99] is defined by a set of assumptions relating all relevant entities in a generally non-deterministic manner (by assigning probabilities to joint occurrences). That is to say, the model embodies assumptions about the hidden mechanics underlying our observations, and we can infer the hidden mechanics within the confines of the model assumptions due to the probabilistic levers connecting unobserved cause and the observed effect. Learning in probabilistic models (*probabilistic inference*) therefore corresponds to a process where a subset of entities defined in the probabilistic model is observed, and subsequently one conducts inference on the conditional probability distribution of unobserved variables of interest; i.e., the information contained in the observations is extracted to inform the hidden parts of the model which have given rise to the observation. For a suitable conceptual grouping of the entities of the

probabilistic model, the process of learning by probabilistic inference can be regarded as the alteration of our prior belief to the posterior belief conditional on our observation, corresponding to the Bayes' theorem (discussed in greater detail in section 2.3). While one will often introduce semantic structure by differentiating between observed data, parameters or latent/hidden nuisance variables, this delineation to an extent is artificial and superimposed. In the Bayesian approach, probabilistic reasoning constitutes the *'daring move that puts causes (observations) and effects (parameters) on the same conceptual level'* [74]. The central role models inhabit for the process of learning and reasoning has been aptly summarized by Ghahramani [100]:

> *Models allow one to make predictions, to understand phenomena, and to quantify, compare and falsify hypotheses. Modelling is also at the core of intelligence. Both artificial and biological systems that exhibit intelligence must be able to make predictions, anticipate outcomes of their actions and update their ability to make predictions in light of new data. It is hard to imagine how a system could do this without building models of the environment that the system interacts with.*

In the context of a probabilistic model, this enables us to reason about the world by conducting inference. As formulated by David Barber [2]:

> *The central paradigm of probabilistic reasoning is to identify all relevant variables $x_1, ..., x_N$ in the environment, and make a probabilistic model $p(x_1, ..., x_N)$ of their interaction. Reasoning (inference) is then performed by introducing evidence that sets variables in known states, and subsequently computing probabilities of interest, conditioned on this evidence.*

As such in its most abstract terms, a probabilistic model is defined by a joint density $p(x_1, ..., x_N)$. For the sake of simplicity in the subsequent discussion we will assume a finite-dimensional setting permitting a representation in terms of a probability mass or probability density functions (for non-parametric models see, e.g., [100, 101, 102]); likewise for the sake of our discussion we by default (and without loss of generality) assume $x_i \in \mathbb{R}$. Assigning probabilities to joint occurrences of variables $p(x_1, ..., x_N)$ implies that the observation of any subset of variables $\{x_i, | i \in \mathbb{I}\}$, $\mathbb{I} \subset \{1, ..., N\}$ generally affects the conditional belief of the remaining unobserved variables. Resolving these conditional distributions corresponds to probabilistic inference and learning[1].

[1] In some cases people delineate conceptually between learning *relevant* model parameters and *inferring* latent model parameters. We do not draw this distinction.

2.1.1 Representation of Probabilistic Models

The necessity to provide abstract and formalized descriptions of probabilistic models has given rise to Probabilistic Graphical Models (PGM) [103], i.e. a formalized description of probabilistic models where structures of complex interdependencies and conditional independence assumptions are represented by means of graphs. While graphical models and corresponding specialized inference algorithms have a rich history, we will refer the reader to the suitable literature for a general discussion [104, 103]. A particular type of PGM expedient for our discussion is given by a Bayesian network which can be represented as a directed acyclic graph (DAG), encoding conditional independence assumptions, i.e., implying a factorization of the joint probability density function:

$$p(x) = \prod_{k=1}^{K} p(x_k|\mathrm{pa}_k) \tag{2.2}$$

Here pa_k defines the set of parent nodes for x_k. Considering our previously introduced joint distribution $p(x_1, ..., x_N)$, we may always introduce a factorization $p(x_1)\, p(x_2|x_1)$ $p(x_3|x_1, x_2) \dots p(x_N|x_1, ..., x_{N-1})$. If represented as a Bayesian network, this would correspond to a fully connected graph with a connection between any arbitrary two nodes. The interesting aspect of a Bayesian network - i.e. conditional independence - is therefore encoded in the *absence* of edges between any two random variables. In such a setting we refer to as *Markov blanket* of x_k the set of variables that fully shields the node from the remaining variables in the model. A more general class of PGMs than Bayesian networks is provided by *factor graphs* [103], i.e., an undirected bipartite graph (variable vertices and factor vertices), similarly defining a factorization of the joint distribution as a product over factors. While A Bayesian network can always be converted to a factor graph representation (a process referred to as moralization), the reverse statement does not hold. Inference algorithms have been developed that seek to exploit a specific structure of the graphical model (e.g., message passing, sum-product algorithm [65]). An interesting type of problem in this context lies in the identification of the structure of graphical models from data [105].

2.1.2 Example : Linear Hidden Markov Model

To remove the discussion from the purely abstract domain, let us consider the following first-order Markovian system generally modeling the time-dynamical evolution of some system (for $t = 1, ..., T$):

$$x_{t+1} = A(\theta)\, x_t \quad + L(\theta)\, \epsilon_t \qquad \epsilon_t \sim \mathcal{N}(0, I) \tag{2.3}$$

$$s_{t+1} = C(\theta)\, x_{t+1} + U(\theta)\, \varepsilon_{t+1} \qquad \varepsilon_{t+1} \sim \mathcal{N}(0, I) \tag{2.4}$$

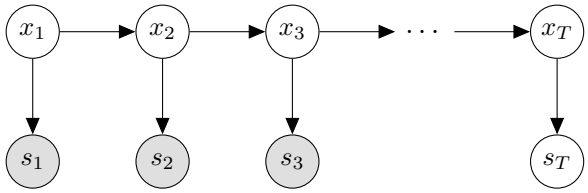


Figure 2.1: Bayesian network representation of first-order hidden Markov model, with observed nodes shaded in gray. The parameters θ are not explicitly depicted in this graphical representation.

Here the matrices $A(\theta) \in \mathbb{R}^{N\times N}$, $C(\theta) \in \mathbb{R}^{M\times N}$, $L \in \mathbb{R}^{N\times N}$ and $U \in \mathbb{R}^{M\times M}$ depend on a set of model parameters $\theta \in \mathbb{R}^{d_\theta}$, which for the time being we assume known. Furthermore we posit $x_1 \sim \mathcal{N}(\mu, \Sigma)$ and assume that $\{s_t | t = 1, ..., T\}$ can be directly observed (in contrast to the latent and unobserved variables $\{x_t | t = 1, ..., T\}$), leading to the so-called *emission* probabilities $p(s_t|x_t, \theta)$ relating unobserved and observed model variables. The special Markovian structure of this model is implied by Eqs. (2.3,2.4) and leads to conditional independence of, e.g., x_t given its Markov blanket $\{x_{t-1}, x_{t+1}, s_t\}$. Furthermore, the Markovian structure is also evident in the factorization of the joint distribution of our hidden Markov Model

$$p(x_{1:T}, s_{1:T} \mid \theta) = p(x_1) \prod_{t=1}^{T} p(x_{t+1}|x_t, \theta) \prod_{t=1}^{T} p(s_t|x_t, \theta) \tag{2.5}$$

, and can also be observed in the representation of this model as a Bayesian network in Figure 2.1. With the probabilistic model fully specified by means of our joint distribution in Eq. (2.5), the process of inference or learning for this model - as previously introduced - corresponds to resolving conditional distributions given observations of a subset of the involved variables. Exemplarily, one might seek to predict the next state in time $p(x_{t+1}|s_{1:t+1}, \theta)$ at time instance t (*filtering*), or instead attempt to infer the latent states given the complete set of observations $s_{1:T}$, i.e., identify the marginal posteriors $p(x_t|s_{1:T}, \theta)$ for $t = 1, ..., T$ (*smoothing*). While under the assumption of Gaussianity and linearity inference on these posterior distributions is attainable in closed form by means of forwards and backward recursion [106, 107], approximate inference methods such as particle filters [108] or variational strategies [109] (as discussed later in section 2.4) have to be employed to approximate the intractable target distributions arising from less restrictive model assumptions. If we consider an extension of the problem where the model parameters θ defining the dynamical behavior of the system also are unknown, this introduces additional complexity and seemingly fundamentally alters the problem. While indeed the necessity to estimate θ alongside the state-trajectory $x_{1:T}$ has far-reaching practical implications, from a sufficiently high and abstract vantage point of probabilistic reasoning the undertaking remains essentially unchanged. Fol-

lowing a Bayesian approach - i.e. treating all unknown model parameters as unknowns - a prior $p(\theta)$ can be introduced, giving rise to a joint distribution $p(x_{1:T}, s_{1:T}, \theta)$ now also encompassing the model parameters θ. In consequence, any conceivable question we wish to answer about this system can again be answered by - generally complex and intractable - conditional distributions of this joint model, such as e.g. making use of the observations $s_{1:T}$ to estimate the system parameters $p(\theta|s_{1:T})$ (with the states $x_{1:T}$ marginalized out), or instead learning jointly both the latent states as well as model parameters, i.e., inferring $p(x_{1:T}, \theta|s_{1:T})$. We provide this example to again echo our previous statements that the general process of probabilistic reasoning and learning is universal and unchanging, and that particular approaches (see, e.g., the Expectation-Maximization algorithm for the identification of θ [110]) merely correspond to *specific* choices and approximations for conducting approximate probabilistic inference (e.g., a flat prior $p(\theta)$ in conjunction with a Dirac approximation for θ). From the viewpoint of probabilistic reasoning, there is no inherent distinction between entities of the model acting semantically as parameters θ or unobserved states $x_{1:T}$ (of course our interests and/or ambitions regarding them might vary). To further emphasize this point, one may observe that reinforcement learning can be interpreted as inference in probabilistic graphical models [71] structurally similar to the hidden Markov model depicted in Fig. 2.1, with now merely different semantic meaning of nodes and edges, and with the observation of $s_{1:T}$ substituted by pseudo-observed optimality variables which we condition on (in such a setting inference of θ would then correspond to finding an optimal policy according to which an actor chooses to influence stochastic state transitions from x_t to x_{t+1}). These examples have been made in the hope to impress upon the reader that within the framework of probabilistic modeling and reasoning, different prediction tasks, seemingly different problem settings (e.g., filtering vs. smoothing) or even entirely different problem domains (e.g., reinforcement learning vs. inference for time-dynamical systems) do not fundamentally differ if viewed from the highest abstraction layer. Learning, prediction and reasoning for any arbitrary system is synonymous with (approximate) probabilistic inference in our probabilistic model, invariably always seeking to resolve distributions conditional on the observed evidence (albeit in possibly extremely complex, challenging and varying settings).

2.2 Information Theory and Statistical Manifolds

In the context of probabilistic reasoning and machine learning, inevitably the question arises of how to quantify information encoded in probability distributions, or how to quantitatively assess the discrepancy of information encoded in two distinct distributions $p(x)$ and $q(x)$ (recall our previous observation that *'probability distributions ... [are] carriers of incomplete information'* [1]). Similarly one would want to be able to assess how informative observation of one random variable x_1 is w.r.t. another *dependent* random variable x_2 in the context of a joint distribution $p(x_1, x_2)$ (which does not factorize). As we shall see, both these questions necessitate scoring the similarity or dissimilarity of two distributions in terms of their distribution of probability mass. To this end let us consider two densities $p, q \in \mathcal{P}$ with $\mathcal{P}$ the set of all admissible probability density functions (with assumed identical support), and introduce a *divergence* measure $D[\cdot||\cdot] : \mathcal{P} \times \mathcal{P} \to \mathbb{R}_0^+$ as any mapping which satisfies:

(i) $D[p||q] \geq 0 \quad \forall \quad p, q \in \mathcal{P}$

(ii) $D[q||p] = 0 \iff q = p$

As such $D[\cdot||\cdot]$ returns a measure of dissimilarity which attains zero value if (and only if) the two distributions are identical ($q = p$). Notably, we do not require $D[\cdot||\cdot]$ to be symmetric nor to satisfy the triangle inequality - as such it is not a proper metric. An example for a divergence measure is given by the Csiszàr's family of f-divergences $D_f[p||q] = \int f(p(x)/q(x))q(x)\,\mathrm{d}x$, with f a suitable[2] convex function [111]. For specific choices of f, we recover important instances such as, e.g., the forward and backward Kullback-Leibler divergence (KLD). Although $D[\cdot||\cdot]$ does not define a proper metric in the general case (due to lack of symmetry and triangle equality), it can nonetheless be regarded as a *statistical distance*, i.e., as quantifying the discrepancy of information encoded in p and q (in terms of the mismatch of probability mass). The most important and widely used divergence measure is given by the previously mentioned Kullback-Leibler divergence [112]

$$D_{KL}[q||p] = \int q(x) \log\left(\frac{q(x)}{p(x)}\right) \mathrm{d}x \tag{2.6}$$

Inspecting Eq. (2.6) we note that this defines an expectation of the log-ratio of densities w.r.t. $q(x)$, and correspondingly the Kullback-Leibler divergence increases with the discrepancy of densities assigned by q and p. Clearly the divergence measure $D[q||p]$ most heavily penalizes cases in which $q(x)$ assigns any significant probability

[2] Convexity is not the only requirement

mass in regions where $p(x)$ approaches zero, leading to mode-snapping behavior of q in the context of variational inference (see upcoming discussion in section 2.4). In contrast, the reverse Kullback-Leibler divergence $D_{KL}[p||q]$ would generally be minimized by a distribution q overestimating the support of p, if zero divergence is not attainable. Within the wide ensemble of possible divergence measures the Kullback-Leibler divergence inhabits a special role, among other reasons because it corresponds to the *relative entropy* (see upcoming section 2.2.2) and relates to the evidence lower bound (see upcoming section 2.4.1.1).

2.2.1 Statistical Manifolds

In the context of a parametric family of distributions $\mathcal{P} = \left\{p(x|\theta)\middle|\theta \in \mathbb{R}^d\right\}$ it follows from our discussion up to this point that similarity or distance of two distributions $p, q \in \mathcal{P}$ should be quantified in terms of the *mismatch of probability mass*. This also implies that the underlying structure of $\mathcal{P}$ is non-euclidian, as we cannot meaningfully gauge the similarity of two distributions $p, q \in \mathcal{P}$ by means of their corresponding parameter values θ_p, θ_q. While we will not attempt any complete or mathematically rigorous discussion, the question about the underlying structure of $\mathcal{P}$ leads to the observation that $\mathcal{P}$ defines a d-dimensional statistical manifold with *non-euclidian* geometry. The study and description of such manifolds by means of differential geometry is referred to as *information geometry* [113, 114, 115], as each point on the manifold corresponds to an encoding of information about x by means of the distribution $p(x|\theta)$. One possible way to obtain the metric tensor of this Riemannian manifold is by considering a second-order Taylor approximation for the previously discussed Kullback-Leibler divergence. If - with slight abuse of notation - we define the distribution $p := p(x|\hat{\theta})$ and an infinitesimal perturbation $q := p(x|\hat{\theta} + \epsilon\theta)$, then a Taylor approximation yields (up to second order terms)

$$\begin{aligned} D_{KL}[q||p] &= \frac{1}{2}\frac{d}{d\epsilon^2}\int\left(p\left(x\middle|\hat{\theta}+\epsilon\theta\right)\log\left(\frac{p\left(x\middle|\hat{\theta}+\epsilon\theta\right)}{p\left(x\middle|\hat{\theta}\right)}\right)\right)\mathrm{d}x \\ &= \frac{1}{2}\epsilon^2\theta^T\mathbb{E}_{p(x|\hat{\theta})}\left[\nabla_\theta \log p\left(x\middle|\hat{\theta}\right)\nabla_\theta \log p\left(x\middle|\hat{\theta}\right)^T\right]\theta \end{aligned} \tag{2.7}$$

where one can identify the symmetric and positive definite *Fisher information matrix* $\mathcal{I}(\theta) \in \mathbb{R}^{d\times d}$ [116, 117] arising as the Riemannian metric tensor of the statistical manifold (at $\hat{\theta}$) induced by our divergence measure[3]

[3]The Fisher information matrix - if not in the context of a metric tensor of statistical manifolds - is generally only positive semi-definite.

$$\mathcal{I}(\theta) = \mathbb{E}_{p(x|\theta)}\left[\nabla_\theta \log p(x|\theta) \nabla_\theta \log p(x|\theta)^T\right] \tag{2.8}$$

The entries of this matrix $\mathcal{I}(\theta)$ quantify the information that x on average conveys regarding the parameter values of θ (correspondingly it appears in the Cramér–Rao bound [118] as a threshold of the attainable precision for unbiased statistical estimators of θ). Fisher information also connects to natural selection and evolutionary dynamics [119], as the genome accumulates information about the environment. The Fisher information matrix as a metric tensor for statistical manifolds is *unique* (up to a constant) and can be derived from other principles as well (e.g., from distinguishability). This uniqueness arises from the necessity of invariance under Markov mappings [120, 115], i.e., information monotonicity [121, 122].

2.2.2 Information Theory

Having remarked upon the non-euclidian nature of statistical manifolds, we will conclude this section by providing brief summaries of the most important information-theoretic quantities as required for our purposes (for simplicity and brevity, assuming continuous and real-valued random variables). The field of information theory was originally pioneered by Shannon [17] in 1948 and studied the problem of communicating over a noisy channel with assumed distributions over the transmitted signal. We refer to the literature for a more in-depth discussion [116, 75].

Entropy The (differential) entropy provides a measure of the uncertainty associated with a random variable $x \sim p(x)$

$$\mathbb{H}[x] = \mathbb{E}_{p(x)}[-\log p(x)] \tag{2.9}$$

and is defined as the expectation of the negative $\log p(x)$, which can be regarded as the *surprisal* of observing x. The entropy, therefore, represents the *average, expected surprisal*. Correspondingly, the entropy collapses to zero for degenerate distributions which place their entire probability mass on a single outcome, and the entropy increases as probabilities are assigned more equally to all possible values of x. If the logarithm is given with respect to base 2, information is measured in bits, whereas for base e we refer to it as nats (this will be our default, unless stated otherwise). Considering e.g. the case of a discrete Binomial random variable $x \sim \mathcal{B}(x|\theta)$ with $\theta \in [0, 1]$, then the entropy (measured in bits) is 0 for $\theta = \{0, 1\}$, and $\mathbb{H}[x] = 1$ for $\theta = 0$. We note that the *differential entropy* can attain negative values and differs in principle from the entropy of discrete random variables (i.e., it is not a perfect equivalence). If x defines a

random vector, or if we consider two random variables x and y, the (joint) entropy $\mathbb{H}[x,y] = \mathbb{E}_{p(x,y)}[-\log p(x,y)]$ follows analogously.

Relative Entropy Given two distributions $q(x)$ and $p(x)$, the relative entropy can be regarded as the *coding inefficiency* that arises from encoding a message under the assumption $x \sim q(x)$, if in actuality $x \sim p(x)$ (i.e., *'expected excess surprisal from using q as a model when the actual distribution is q'* [123]). The relative entropy equates to the previously introduced Kullback-Leibler divergence

$$D_{KL}[\, p(x)||q(x)\,] = E_{p(x)}\left[\log\left(\frac{p(x)}{q(x)}\right)\right] \tag{2.10}$$

$$= \mathbb{E}_{p(x)}[-\log q(x)] - \mathbb{H}[p(x)] \tag{2.11}$$

and in Eq. (2.11) has been expressed with respect to the *cross-entropy* $\mathbb{H}[q,p] = \mathbb{E}_{p(x)}[-\log q(x)]$ and the entropy of $p(x)$. While (for base 2) the relative entropy defines the average number of *additional* bits required to encode x, the relationship defined by Eq. (2.11) suggests that cross-entropy corresponds to the *total* number of bits required to encode $x \sim p(x)$ under the sub-optimal assumption $x \sim q(x)$. Classical loss functions (e.g. discriminative classification problems) in machine learning often entail cross-entropy, as it defines the only non-constant term of the Kullback-Leibler divergence in the context of fixed empirical data distributions.

Conditional Entropy Given a joint distribution $p(x,y)$, the conditional entropy quantifies the remaining uncertainty in y conditionally on having observed x, and is defined as the expectation of the entropy of the conditional distribution $p(y|x)$ with respect to $p(x)$

$$\mathbb{H}[y|x] = \mathbb{E}_{p(x)}\left[\mathbb{E}_{p(y|x)}[-\log p(y|x)]\right] \tag{2.12}$$

$$= -\mathbb{E}_{p(x,y)}\left[\log\left(\frac{p(x,y)}{p(x)}\right)\right] \tag{2.13}$$

From the definition of the entropy and conditional entropy one can derive the chain rule, according to which the (joint) entropy of a random vector $[x,y]$ can - rather intuitively - be expressed and decomposed in terms of marginal entropies and conditional entropies [116]

$$\mathbb{H}[x,y] = \mathbb{H}[x] + \mathbb{H}[y|x] \tag{2.14}$$

$$= \mathbb{H}[y] + \mathbb{H}[x|y] \tag{2.15}$$

If x and y are not independent, the conditional distributions $p(y|x)$ and $p(x|y)$ will see a reduction in entropy and thus uncertainty upon observing either x or y. Consequently, the conditional entropy needs to relate to the amount of information that x conveys about y or vice versa (see upcoming Eqs. (2.17 - 2.20)), given the dual nature of information and uncertainty.

Mutual Information We again consider two random variables x, y with a joint distribution $p(x, y)$. The mutual information $\mathbb{I}(x, y)$ quantifies the dependence of two random variables as is defined as the Kullback-Leibler divergence of the joint and the product of marginal probability density functions

$$\begin{aligned}\mathbb{I}[x, y] &= \int p(x, y) \log\left(\frac{p(x, y)}{p(x)\, p(y)}\right) \mathrm{d}x\, \mathrm{d}y \\ &= D_{KL}[p(x, y)||p(x)\, p(y)]\end{aligned} \tag{2.16}$$

The properties of the KLD imply that the mutual information $\mathbb{I}[x, y]$ must be both *non-negative* and *symmetric*, only attaining zero value for independence of x and y (implying a factorization of the density $p(x, y) = p(x)\, q(x)$). In contrast to, e.g., the correlation of x and y, the mutual information can also capture complex non-linear dependence. Furthermore, if the mutual information quantifies the amount of information gained upon observing one of the two random variables (in, e.g., bits or nats), there necessarily must exist a relationship to the previously introduced conditional entropies, which can be derived as [124]:

$$\mathbb{I}[x, y] = \mathbb{H}[y] - \mathbb{H}[y|x] \tag{2.17}$$
$$= \mathbb{H}[x] - \mathbb{H}[x|y] \tag{2.18}$$
$$= \mathbb{H}[x] + \mathbb{H}[y] - \mathbb{H}[x, y] \tag{2.19}$$
$$= \mathbb{H}[x, y] - \mathbb{H}[x|y] - \mathbb{H}[y|x] \tag{2.20}$$

I.e., mutual information corresponds to the reduction of marginal entropy upon observing a dependent random variable. Unsurprisingly, the concept of mutual information is relevant when trying to extract informative and compressed encodings of data, for instance in the context of an information bottleneck [125] (other machine learning techniques explicitly seeking to exploit mutual information exist, e.g., [126, 127]).

2.3 Bayes' Theorem

2.3.1 Updating States of Belief

If the Bayesian viewpoint as previously stated proclaims that *'probability is a measure of the degree of belief about a proposition'*, then Bayes' theorem constitutes the *'rule for manipulating states of belief'* [84] in light of new information or data. Alternatively, one may consider Bayes' theorem as a mechanism by which (possibly incrementally and recursively) information is accumulated, with information - as previously already remarked - being dual to the *'resolution of uncertainty'* [17]. While Bayes' theorem trivially follows from the definition of conditional probabilities or conditional densities, it is difficult to overstate its importance, as it equates to the process of inference at the heart of probabilistic reasoning and machine learning. For this reason, we will discuss some non-obvious implications and properties implicitly defined by Bayes' theorem. To this end, we again assume a parametric, finite-dimensional probabilistic model that permits a representation via a probability density function $p(x|\theta)$ with real-valued vectors x and θ [4]. Let us furthermore assume that there exists a finite set of competing hypothesis $\mathcal{H}_i, i \in 1, ...I$, implying different possible explanations of the data and as such different probabilistic models $p(x|\theta, \mathcal{H}_i)$. It is our intention to make statements about the plausibility of parameters and competing hypotheses given the observation of a dataset $\mathcal{D} = \{x_n\}_{n=1}^N$. In contrast to the maximum likelihood approach, which seeks to identify the parameters $\hat{\theta}$ that satisfy $\hat{\theta} = \arg\max_\theta p(\mathcal{D}|\theta, \mathcal{H}_i)$, Bayes' theorem states that given a *prior* belief $p(\theta|\mathcal{H})$ the observation of data $\mathcal{D} = \{x_n\}_{n=1}^N$ affects a change in our (prior) probabilistic belief giving rise to the *posterior*

Bayes' theorem

$$p(\theta|\mathcal{D}, \mathcal{H}_i) = \frac{p(\mathcal{D}|\theta, \mathcal{H}_i)\ p(\theta|\mathcal{H}_i)}{p(\mathcal{D}|\mathcal{H}_i)} \tag{2.21}$$

, i.e., information contained within the data $\mathcal{D}$ is consistently combined with a priori available information encoded in the prior distribution $p(\theta|\mathcal{H}_i)$. This of course is contingent on a *model* underlying the likelihood $p(\mathcal{D}||\theta, \mathcal{H}_i)$, i.e., the ability to make statement about the data expected to be observed conditional on θ and $\mathcal{H}_i$. As Bayes' theorem permits to relate forward predictions and forward models with the plausibility of parameters θ conditional on the observed data, it is also sometimes referred to as the inverse law of probabilities. In consequence, the posterior $p(\theta|\mathcal{D}, \mathcal{H}_i)$ defines the

[4] The parameters $\theta \in \Theta$ and data $x \in \mathcal{X}$ can correspond to - in principle - arbitrarily simple entities, or alternative e.g. the set of all symmetric positive definite matrices (manifold), or Hilbert function spaces.

solution for the probabilistic treatment of any inverse problems, with the forward model defining the predictions in the likelihood term [93, 128]. In Bayes' theorem (2.21), all involved densities are assigned different names according to the role they take in this context:

Prior The prior distribution $p(\theta|\mathcal{H}_i)$ defines the a priori belief regarding the parameters θ of the model $\mathcal{H}_i$, i.e., *prior* to observing the data $\mathcal{D}$. The prior may either incorporate a priori available information (as well as encode desirable model bias in the context of machine learning), but can also be constructed to remain uninformative in any setting where this is not applicable (see section 2.3.4.)

Likelihood The likelihood $p(\mathcal{D}|\theta, \mathcal{H}_i)$ is a distribution in the *data* and defines the probability (or the probability density) of observing a specific dataset $\mathcal{D}$ *conditional* on the model parameters θ and hypothesis $\mathcal{H}_i$. In conjunction with the prior the likelihood scores plausibility of the parameters θ by their ability to explain the observed data $\mathcal{D}$.

Evidence The evidence term $p(\mathcal{D}|\mathcal{H}_i)$ acts as a normalizing term and quantifies the marginal probability or probability density of observing the dataset $\mathcal{D}$ given hypothesis $\mathcal{H}_i$, therefore relating to the plausibility of $\mathcal{H}_i$. Note that the evidence term corresponds to a marginal density $p(\mathcal{D}|\mathcal{H}_i) = \int p(\mathcal{D}|\theta, \mathcal{H}_i)\, p(\theta|\mathcal{H}_i)\, \mathrm{d}\theta$ which is intractable in almost all interesting problem settings.

Failure to account for the prior information, i.e. conflating the plausibility of parameters given the data with the probability of the data given the parameters, is in some settings known as the base rate fallacy (consider exemplarily the prevalence of false positive tests). For a *flat* prior $p(\theta|\mathcal{H}_i) \propto$ const. this distinction disappears, and the maximum likelihood estimate and maximum a posteriori (MAP) estimate $\hat{\theta} = \arg\max_\theta p(\theta|\mathcal{D}, \mathcal{H}_i)$ will coincide - but as to be discussed later, a flat prior does not necessarily correspond to a uninformative prior (section 2.3.4). One can also show that Bayes' theorem arising from our framework of probabilistic reasoning satisfies [129, 130] for any arbitrary possible PDF $\rho(\theta)$ the convex optimality criterion [131]

$$p(\theta|\mathcal{D}, \mathcal{H}_i) = \min_{\rho(\theta)} \left\{ D_{KL}\left[\rho(\theta)||p(\theta|\mathcal{H}_i)\right] - \mathbb{E}_{\rho(\theta)}\left[\log p(\mathcal{D}|\theta, \mathcal{H}_i)\right] \right\} \tag{2.22}$$

which identifies the posterior as the density $\rho(\theta)$ maximizing the marginal probability of observing the data $\mathbb{E}_{\rho(\theta)}\left[\log p(\mathcal{D}|\theta, \mathcal{H}_i)\right]$, while also balancing it against the shift from the prior belief in terms of relative entropy (i.e., the information-distance implied

by the shift of prior distribution to $\rho(\theta)$). In this context we mention that similarly to Eq. (2.22) the maximum likelihood estimate (MLE) $\hat{\theta}$ can also be phrased as the minimization of the Kullback-Leibler divergence with respect to the *empiricial distribution* $p_e(x) = (1/N)\sum_{n=1}^{N}\delta(x - x_n)$ defined by the data $\mathcal{D}$, i.e. $\hat{\theta} = \arg\min D_{KL}[p_e(x)||p(x|\theta)]$. As we are only able to inform model predictions by means of our inferred parameters θ, Ghahramani observed that *'the parameters in parametric models constitute a bottleneck in this information channel from past data $\mathcal{D}$ to future predictions x'* [100][5]. This contrasts with nonparametric methods, where the complexity of the model adapts automatically to the data due to the infinite-dimensional nature of the parameters [102, 101]. Within the confines of parametric models, techniques like sparsity-inducing priors and automatic relevance determination [132, 133, 134] can be used to incentivize the model to make only partial use of the information channels available (based on the inherent complexity of the data or some desired *structured* and *interpretable* explanation of the observed data). Similarly, hierarchical probabilistic models and hyperpriors can be adopted when following a fully Bayesian approach [135, 136], i.e., when treating all unknown parameters probabilistically. Both the evidence term as well as the posterior are generally not tractable, with the exception of narrowly constrained conjugate exponential models [65]. As such, specialized algorithms have to be employed which seek to provide a *numerical approximation* of the posterior (see the ensuing discussion in section 2.4). In passing we remark that given suitable inference methods such as reversible jump Markov Chain Monte Carlo [137] one may also consider parametric models where the number of model parameters itself is variable (e.g., Bayesian DNA sequence segmentation [138]). Another feature of Bayes' theorem is that it straightforwardly lends itself to sequential and iterative updating, i.e., the sequential arrival of data $\mathcal{D} = \cup_{k=1}^{K}\mathcal{D}_k$ as an ensemble of K batches $\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_K$ can be phrased in terms of recursive updates $p(\theta|\mathcal{D}_1, ..., \mathcal{D}_k, \mathcal{H}_i) \propto p(\mathcal{D}_k|\theta, \mathcal{H}_i)\, p(\theta|\mathcal{D}_1, ..., \mathcal{D}_{k-1}, \mathcal{H}_i)$. Here the posterior recursively acts as the new prior the given the latest batch of observed data. This type of iterative updating arises naturally in a setting where data arrives sequentially, as well as of course for time-dependent and time-dynamic inference problems [139]. Traditional deterministic, computational methods can also be subsumed into Bayes' theorem from the vantage point of probabilistic numerics [8, 10, 9, 12], which phrases the solution of numerical problems as an inverse inference problem (uncertainty, e.g., being introduced by finite machine precision or the constraints of tempo-spatial discretization). Another recent trend has been the emergence of probabilistic programming and associated numerical frameworks [140, 141, 142, 143, 144], which seek to provide a more automated construction of probabilistic models and execution of (approximate) probabilistic inference.

[5] Hence why we will explore and exploit the notion of effective physical properties as very narrow information channels enabling predictions in the small data domain, see section 3.1

2.3.2 Posterior Predictive and the Bayes Estimator

The posterior distribution $p(\theta|\mathcal{D}, \mathcal{H}_i)$ represents the combined information of the prior and the data $\mathcal{D}$ conditional on our model assumptions $\mathcal{H}_i$. After obtaining the posterior distribution (or a sufficiently accurate approximation), one may employ it to make predictions w.r.t. x conditional on our model (i.e., hypothesis $\mathcal{H}_i$) by means of the *posterior predictive*

$$p(x|\mathcal{D}, \mathcal{H}_i) = \int p(x|\theta, \mathcal{H}_i)\, p(\theta|\mathcal{D}, \mathcal{H}_i)\, \mathrm{d}\theta \tag{2.23}$$

which importantly marginalizes out the epistemic uncertainty in our model parameters θ. Obtaining the posterior and posterior predictive translates to the objective in the engineering domain to make predictions or to make decisions and/or inform competing designs. The distributions $p(x|\mathcal{D}, \mathcal{H}_i)$ and $p(\theta|\mathcal{D}, \mathcal{H}_i)$ capture all the relevant information available about the model and model predictions given the data $\mathcal{D}$. In particular one may in principle evaluate any arbitrary expectation under the posterior predictive

$$\mathbb{E}_{p(x|\mathcal{D},\mathcal{H}_i)}\left[f(x)\right] = \int f(x)\, p(x|\mathcal{D}, \mathcal{H}_i)\, \mathrm{d}x \tag{2.24}$$

If one seeks to reduce θ to a point estimate, a principled way to do so is by means of the Bayes estimator $\tilde{\theta}$, which minimizes the posterior expectation of a suitable *risk* function or *utility* function. I.e., select as Bayes estimator the parameters $\tilde{\theta}$ which minimize the a posteriori expected Bayes risk $\arg\max_{\tilde{\theta}} \mathbb{E}_{p(\theta|\mathcal{D},\mathcal{H}_i)}[L(\theta, \tilde{\theta})]$, with $L(\theta, \tilde{\theta})$ a suitable application-specific loss function depending on the parameters.

2.3.3 Bayes Factor

If our preceding discussion has identified the correct way to consistently update our states of belief regarding the parameters θ conditional on the hypothesis $\mathcal{H}_i$, the question remains how plausible or credible the competing hypotheses $\mathcal{H}_i$ are in light of the observed data $\mathcal{D}$. The evidence ratio, i.e., the relative probability of observing the data under different hypotheses, in combination with our a priori belief $p(\mathcal{H}_i)$ regarding model plausibility gives rise to the *Bayes Factor* [145]

$$BF_{ij} = \frac{p(\mathcal{H}_i|\mathcal{D})}{p(\mathcal{H}_j|\mathcal{D})} = \frac{p(\mathcal{D}|\mathcal{H}_i)}{p(\mathcal{D}|\mathcal{H}_j)} \frac{p(\mathcal{H}_i)}{p(\mathcal{H}_j)} \tag{2.25}$$

Bayes Factor BF_{ij}	Interpretation
> 100	Extreme evidence for $\mathcal{H}_i$.
$30 - 100$	Very strong evidence for $\mathcal{H}_i$.
$10 - 30$	Strong evidence for $\mathcal{H}_i$
$3 - 10$	Moderate evidence for $\mathcal{H}_i$
$1 - 3$	Anecdotal evidence for $\mathcal{H}_i$
1	No evidence

Table 2.1: Jeffrey's scale of the Bayes factor as given in [147], providing the relative weight of evidence of the data for $\mathcal{H}_i$ over $\mathcal{H}_j$. Relative plausibilities of $BF_{ij} < 1$ follow from symmetry.

as the relative plausibility (or posterior odds) of two competing hypotheses $\mathcal{H}_i$ and $\mathcal{H}_j$ upon observing the data $\mathcal{D}$. Notably we do not require the marginal probability $p(\mathcal{D})$, which cancels out. The Bayes factor BF_{ij} depends on the probability of observing the data conditional on the hypothesis, i.e., involves the generally intractable marginal distribution

$$p(\mathcal{D}|\mathcal{H}_i) = \int p(\mathcal{D}|\theta, \mathcal{H}_i)\, p(\theta|\mathcal{H}_i)\, \mathrm{d}\theta \tag{2.26}$$

While we will defer discussion on the underlying mechanism by which Bayes' theorem and thus BF_{ij} give preference to competing hypotheses (section 2.3.5), a comparably simple way to at least interpret the Bayes factor is by considering it as *betting odds*, i.e., expressing to which extent the data $\mathcal{D}$ favors hypothesis $\mathcal{H}_i$ over $\mathcal{H}_j$ (or the other way around). It is therefore obvious that the plausibility of a model or hypothesis $\mathcal{H}_i$ regarding the observation of some data $\mathcal{D}$ can only be quantified in *relative terms* (compared to another competing explanation $\mathcal{H}_j$), and not in *absolute* terms (see also Bayesian hypothesis testing [146]). A qualitative interpretation of the Bayes factor - known as the Jeffreys scale - has been given in Table 2.1.

2.3.4 Uninformative and Improper Priors

Based on our preceding discussion of the non-euclidean structure of probability distributions as well as from Bayes' theorem and its parametrization dependence it follows that the notion of an *uninformative* prior must be different from the notion of a *flat* prior $p(\theta|\mathcal{H}_i) \propto$ const., which would arise from the naive extension of the *principle of insufficient reason* [148] in a discrete setting. An uninformative prior can rather be understood as assigning a priori *'equal probability to equal volumes of the statistical manifold'* [149], or alternatively, to equal volumes of the hypothesis space. One can show that for a parametric statistical manifold with the Fisher information matrix $\mathcal{I}(\theta)$

as the Riemannian metric tensor (see previous discussion in section 2.2.1), the notion of equal probability for equal volumes in the hypothesis space leads to *Jeffreys prior* [150]

$$p(\theta) \propto \sqrt{\det \mathcal{I}(\theta)} \tag{2.27}$$

While this often constitutes an improper prior (i.e., not integrating to a finite value), the associated posterior may still generally define a valid distribution. Despite the theoretical justification and elegance of Jeffreys prior, from a pragmatic perspective it has been known to yield suboptimal results in particular in a high-dimensional setting (even assuming that $\mathcal{I}(\theta)$ is attainable). This observation gave rise to the proposition and development of *reference priors* [151, 152], which seek to identify an uninformative prior distribution $p(\theta)$ by maximizing the *expected* information distance between prior and posterior (a concept also explored in the context of deep learning [153]). Importantly, the reference prior is not defined with respect to a specific observed dataset, but instead implies an expectation w.r.t. the data predicted to be observed by the model. For certain simple cases, Jeffreys' prior coincides with the reference prior [152].

2.3.5 Occam's razor and Geometric Complexity

Bayes' theorem inherently trades off model complexity and model fit giving preference to simpler explanations of the data - a general principle known as *Occam's razor*. This is reflected already in Eq. (2.22), if we regard as a measure of complexity the extent to which the probability mass of the posterior has shifted from the prior distribution (as measured by the Kullback-Leibler divergence). Notably, the preference for simplicity is not postulated or artificially introduced, but rather arises naturally from the basic rules of probability theory (and therefore Cox's axioms) [132]. From a machine learning perspective, one can remark that the ability to increase the model fit to the observed data does not equal a better generalization performance (see also bias-variance decomposition [65]). The most accessible and most widely stated explanation as to why Bayes' theorem incorporates Occam's razor is given by the necessity of the model to define a *normalized* distribution $p(\mathcal{D}|\mathcal{H}_i)$ in the data. This implies that a more complex model - which may account and explain for the observations of a greater variety of data - therefore necessarily must generally assign a smaller probability (or probability density) $p(\mathcal{D}|\mathcal{H}_i)$ compared to a simpler model, as long as both models are able to provide an explanation for the observed data. This mechanism is illustrated in Figure 2.2 reproduced from MacKay [75], where the conceptual idea is visualized for a simplistic model. We assume the existence of three hypotheses $i = 1, 2, 3$, corresponding to an approximately uniform distribution in x with its spread depending on the variance of the parameter θ (consider *conceptually* a model akin to $x|\theta \sim \mathcal{N}(\theta, 1), \theta|\mathcal{H}_i \sim \mathcal{N}(0, \sigma_i^2)$).

If one assumes that the posterior $p(\theta|\mathcal{D}, \mathcal{H}_i)$ is sufficiently strongly peaked around the maximum a posteriori estimate of θ, using Laplace's method one can approximate the evidence term as [75]:

$$\begin{aligned} p(\mathcal{D}|\mathcal{H}_i) &= \int p(\mathcal{D}|\theta, \mathcal{H}_i)\, p(\theta|\mathcal{H}_i)\, \mathrm{d}\theta \\ &\approx \underbrace{p(\mathcal{D}|\theta_{\mathrm{MAP}}, \mathcal{H}_i)}_{\text{Achieved Likelihood}} \cdot \underbrace{\frac{\sigma_{\theta|\mathcal{D}}}{\sigma_\theta}}_{\text{Occam factor}} \end{aligned} \tag{2.28}$$

The values of $\sigma_{\theta|\mathcal{D}}$ and σ_θ characterize the variance of unimodal prior and posterior distributions in θ, respectively (see also Figure 2.2). This suggests that the model evidence can be interpreted as the achieved likelihood penalized by the *Occam factor*, which MacKay characterized as *'the ratio of the posterior accessible volume of parameter space $\mathcal{H}_i$'s to the prior accessible volume'* [154]; i.e. the complexity of a hypothesis is measured to the extent it requires to shift the prior's probability mass. An extension of this discussion and analysis has been pursued by Balasubramanian from an information geometric perspective [155], considering competing models with d-dimensional bounded parameter space ($\theta \in \Theta, \Theta \subset \mathbb{R}^d$) and non-informative Jeffreys prior $p(\theta) \propto \sqrt{\det \mathcal{I}(\theta)}$. If we introduce the negative model evidence $\mathcal{X} = -\log p(\mathcal{H}|\mathcal{D})$ as a measure of the complexity of our model $\mathcal{H}$, then similarly to our previous discussion given a sufficiently large dataset $\mathcal{D} = \{x_n\}_{n=1}^N$ resulting in a sharply peaked unimodal posterior enables approximation of the complexity term $\mathcal{X}$ as [155, 156]

$$\mathcal{X} \approx -\log p\left(\mathcal{D}|\hat{\theta}\right) + \log\left(\frac{V(\mathcal{H})}{V_c(\mathcal{H})}\right) + \mathcal{O}(1/N) \tag{2.29}$$

The ratio of V and V_c again corresponds to the previously mentioned ratio of prior and posterior accessible *volume* of the parameter space in Θ, i.e., $V_c(\mathcal{H})$ constitutes the volume of the subspace of Θ which yields non-negligible probability for the observed data $\mathcal{D}$ (this is basically a generalization of Eq. (2.28)). Additionally, it was shown by Balasubramanian that for suitable simplifying assumptions (restricting contributions to the integrals to the vicinity of the maximum likelihood estimate $\hat{\theta}$) the measure of complexity $\mathcal{X}$ of the model can be expressed and further decomposed in terms of *stochastic* and *geometric complexity* [6] [155]. Let $\hat{\theta}$ denote the maximum likelihood estimate of the parameters, then approximately:

[6] Note that we merely reproduce the final results, and the reader is referred to the original publication for detailed discussion and derivation.

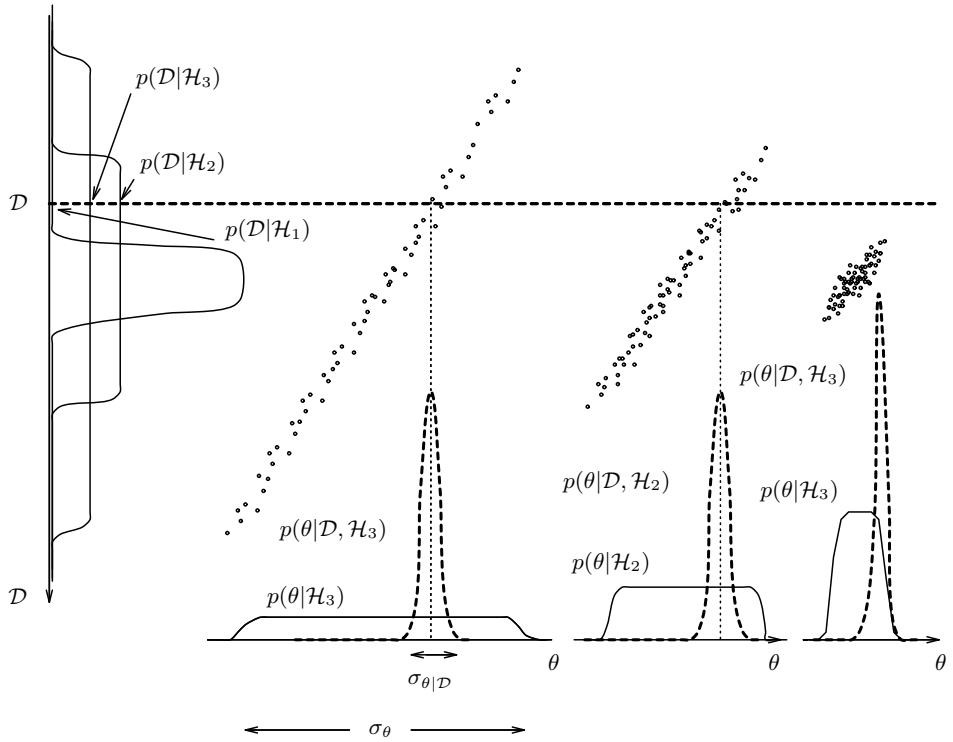


Figure 2.2: Illustration of Occam's razor as given by MacKay (Figure 28.6 in [75], with slight adaptions to notation). Our three different models $\mathcal{H}_1, \mathcal{H}_2$ and $\mathcal{H}_3$ correspond to three approximately uniform distribution in x with the spread depending on the variance of the parameter θ (as stated in the main text, *conceptually* a model akin to $x|\theta \sim \mathcal{N}(\theta, 1), \theta|\mathcal{H}_i \sim \mathcal{N}(0, \sigma_i^2)$); for $i = 1, 2, 3$ we obtain different variability in $p(\theta|\mathcal{H}_i)$ and $p(\mathcal{D}|\mathcal{H}_i)$. We compare the relative plausibility of these three models under the assumption of having observed a *single* datapoint $\mathcal{D}$. Following Eq. (2.28), the most plausible hypothesis is $\mathcal{H}_2$, as it defines the simplest model which nonetheless still is able to explain the observed data (i.e., the model $\mathcal{H}_2$ can assign a larger density to the data $\mathcal{D}$ compared to $\mathcal{H}_3$, due to its smaller support). *Reproduced with permission of The Licensor through PLSclear.*

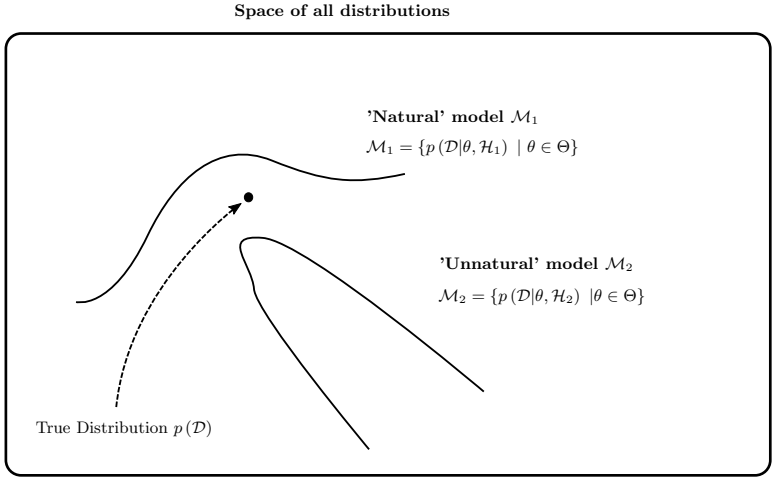


Figure 2.3: Illustration of geometric formulation of Occam's razor for parametric models based on Figure 2 by Balasubramanian [157]. While both natural $\mathcal{M}_1$ and unnatural model $\mathcal{M}_2$ can be identically close to the ground truth for suitable parameter values $\hat{\theta}$, the unnatural model scores poorly on geometric complexity (as small deviations lead to rapid deterioration of its ability to explain the data); furthermore it needs to be assumed that a different realization of the data $\mathcal{D}$ could strongly affect the estimate of θ. Hence the choice according to Bayesian evidence favors the natural and less complex model $\mathcal{M}_1$.

$$\mathcal{X} \approx -\log p\left(\mathcal{D}\middle|\hat{\theta}\right) + \frac{d}{2}\log\frac{N}{2\pi} + \log\int\sqrt{\det\mathcal{I}(\theta)}\,\mathrm{d}\theta + \frac{1}{2}\log\left(\frac{\det J\left(\hat{\theta}\right)}{\det\mathcal{I}\left(\hat{\theta}\right)}\right) + \mathcal{O}(1/N) \tag{2.30}$$

Here $\mathcal{I}(\hat{\theta})$ and $J(\hat{\theta})$ denote the Fisher information matrix as well as a kind of empirical Fisher information matrix $[J(\hat{\theta})]_{uv} = -(1/N)\nabla_{\theta_u}\nabla_{\theta_v}\log p(\mathcal{D}|\theta)|_{\hat{\theta}}$ around the maximum likelihood estimate $\hat{\theta}$ (normalized by the number of datapoints N). The expression in Eq. (2.3) can again be viewed to relate to the achieved log-likelihood penalized by the model complexity. The first two terms of $\mathcal{X}$ in this expansion define the *stochastic complexity* and relate to the *minimum description length* (MDL) principle [158], i.e., favor a hypothesis $\mathcal{H}$ that offers a parsimonious *compression* of the data (while taking into consideration the likelihood achieved for the data). The stochastic complexity scales with the number of data points as $\mathcal{O}(N)$ and $\mathcal{O}(\log N)$. The Bayesian approach of model selection can therefore be seen to extend upon the stochastic complexity criterion of the minimum description length, adding terms that capture the *robustness* of the proposed model in the hypothesis space. The third term in Eq. (2.3) essentially provides a measure of the prior support of the model, and as

such penalizes models to the degree they are a priori unconstrained. We can also note the second and third terms of our expression (2.3) for the complexity $\mathcal{X}$ do not depend on the observed data $\mathcal{D}$, and as such define an inherent property of the assumed model. In contrast, the last term relates to the Fisher information matrix at the maximum likelihood estimate $\hat{\theta}$ and thereby to the notion of geometric robustness. If the volume spanned by the eigenvectors of the Fisher information matrix $\mathcal{I}(\hat{\theta})$ is small (in relation to $J(\hat{\theta})$), this implies that the model is sensitive with respect to the choice of θ, i.e., perturbation of the parameters away from $\hat{\theta}$ quickly degrade the ability of the model to explain the data $\mathcal{D}$. This is visually illustrated in Figure 2.3, which contrasts a 'natural' model and an 'unnatural model'; while achieving the same likelihood, they score differently according to their geometric robustness. We also note that of course, the discussion of Occam's razor and the Occam factor is theoretical in nature, as the evidence $p\left(\mathcal{D}|\mathcal{H}_i\right)$ remains intractable in the general setting. The purpose of this section is merely to shed light on the mechanism by which the evidence gives preference to models, and how this relates to robustness and generalization. Approximations of the evidence term exist which introduce varying degrees of simplifying assumptions (e.g., based on a Laplace approximation [2], Akaike Information Criterion (AIC) [159], Bayesian Information Criterion (BIC) [160]). Intuitive examples for the preference of the Bayesian evidence towards simpler models may also be found in Tenenbaum's thesis on Bayesian concept learning [161].

2.4 Approximate Probabilistic Inference

Inference pertains to the process of resolving probability distributions conditional on a set of known or observed entities, or more generally, the resolution of any distribution of interest by means of suitable numerical methods. While a large ensemble of algorithms exists which have been tailored to specific models enjoying a specific structure (e.g., factorization, conjugate models), in the following we will confine our discussion to inference methods that have general applicability and as such can target in principle any intractable distribution $p(x)$ without any further restrictive assumptions. Exemplarily, our target distribution p may arise as the posterior $p(x|\mathcal{D})$ upon the observation of a dataset $\mathcal{D}$

$$p(x|\mathcal{D}) = \frac{p(\mathcal{D}|x)\,p(x)}{p(\mathcal{D})} = \frac{\pi(x)}{p(\mathcal{D})} \tag{2.31}$$

with $\pi(x)$ the unnormalized posterior, i.e., $\pi(x) \propto p(x|\mathcal{D})$. Generally, inference necessitates working with the unnormalized distribution $\pi(x)$ due to the intractability of the evidence. We have slightly changed the notation away from θ, as we do not necessarily assume parametric models. Given the vast amount of effort this topic has received, the following exposition by necessity is limited to an introductory discussion of two of the most widely used and most general approaches (variational inference and sampling-based methods). The reader is referred to the appropriate literature for a more in-depth discussion (e.g., [162, 163, 164, 165, 166]). In addition, we should note that the delineation suggested in the following subsections is not absolute, as hybrid variants exist that seek to, e.g., bridge variational inference and sampling-based approaches [167, 168, 169]. Naturally more simplistic approaches can be used which reduce the a posteriori probability mass to a mere point-estimate (*maximum-a-posteriori*), or try to capture the distribution of probability mass with a Gaussian at the mode of the posterior (*Laplace* approximation).

2.4.1 Variational Inference

Variational Inference (VI) [162] rephrases the problem of capturing the probability mass of the posterior p as a variational problem. To this end, we posit that the intractable target distribution can be sufficiently approximated by a distribution q^* within an a priori defined family of distributions Q. We then identify the optimal distribution $q^* \in \mathcal{Q}$ within this family as the minimizer of a divergence metric $D[\cdot||\cdot]$, i.e.

$$q^* = \arg\min_{q \in \mathcal{Q}} D[q||p] \tag{2.32}$$

The variational approach is inherently approximate (unless $p \in \mathcal{Q}$), and the choice of $\mathcal{Q}$ defines a tradeoff between computational complexity and the extent to which q^* is able to capture the probability mass of the target distribution. The divergence measure $D\left[\cdot||\cdot\right]$ of course implicitly defines a mechanism that quantifies the misalignment of probability mass between q and p in a particular way, and hence even for the identical variational family Q, different choices of the divergence metric may define widely different optima $q^* \in \mathcal{Q}$ [7] (e.g., α-divergence [171], Rényi divergence [172], χ-divergence [173], Wasserstein distance [174, 175]). Conditional on the specific choice of $\mathcal{Q}$ and the divergence measure (as well as other practical considerations), a large ensemble of different variational inference methods arise (e.g., [176, 177, 178, 179]). In the general case, we can define $\mathcal{Q}$ irrespective of the specifics of the probabilistic model (*black-box variational inference*), as long as we are able to evaluate the log-joint corresponding to the unnormalized density $\pi\left(x\right)$ [180, 181, 182]. Most commonly we parametrize $\mathcal{Q}_\xi$ by means of variational parameters ξ and subsequently rephrase Eq. (2.32) as a minimization problem w.r.t. the variational parameters, i.e., $\xi^* = \arg\min_\xi D\left[q_\xi||p\right]$. As an example one might seek to approximate p by means of a Gaussian $\mathcal{N}\left(x|\mu,\Sigma\right)$, implying that $\xi = \{\mu,\Sigma\}$ (or a low-rank representation thereof [183] for scaleability) would be a possible choice for the associated variational parameters ξ. Given this example, it should be noted that variational inference is not restricted to families of distribution $\mathcal{Q}$ which permit explicit representation, see e.g. [184, 185, 186]. From an information-geometric perspective considering $\mathcal{Q}_\xi \subset \mathcal{M}$ as a statistical manifold, identifying q^* according to Eq. (2.32) defines an *information projection* of $p \in \mathcal{M}$ onto the submanifold $\mathcal{Q}_\xi$ along the geodesic implicitly defined by $D\left[\cdot||\cdot\right]$ [170], where $D\left[q||p\right]$ can be interpreted as the loss of information incurred by substituting the intractable target distribution p with the approximation q. This geodesic projection of the posterior p is illustrated in Figure 2.4 - with the variational submanifold $\mathcal{Q}_\xi$ (here $\xi \in \mathbb{R}^2$) embedded in the space of all admissible density functions $\mathcal{M}$. While we will not attempt a taxonomy of the various variational inference methods developed, we make mention of a subclass of algorithms (due to their importance) summarized as *Stochastic Variational Inference* (SVI, e.g. [163, 187, 188]). For these algorithms, the divergence (or some proxy) cannot be calculated, and instead, a Monte Carlo estimate of the divergence as well as most often

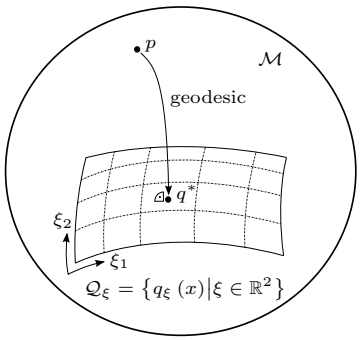


Figure 2.4: Variational inference as projection of $p \in \mathcal{M}$ along a geodesic on variational submanifold $\mathcal{Q} \subset \mathcal{M}$. Figure based on and adapted from [170].

[7] Consider exemplarily the impact of forward and reverse Kullback-Leibler divergence in under- or overestimating the support of the target distribution [65]

also its gradient is used in combination with stochastic optimization techniques (e.g., stochastic gradient ascent [189, 190, 191]). The nature of the Monte Carlo estimator used [192] will generally heavily impact the variance, and hence the viability and scalability of this approach; in consequence, a significant research effort has been aimed at variance reduction of Monte Carlo gradient estimators in this context (e.g. natural gradients [193], path derivative gradient estimator [193], generalized reparametrization gradient [194]). Due to its scalability, SVI is also the most commonly used inference approach for complex, deep probabilistic models [195]; in this setting, SVI can also (additionally) involve subsampling of the data. The approximative nature of variational inference however also runs the risk of a *'self-fulfilling prophecy'* [196], where incorrect and faulty model assumptions are reinforced in the results. Echoing our previous claims that learning is synonymous with probabilistic inference [197], we mention that variational inference has been related to general human cognition and action planning by means of the Bayesian brain hypothesis and the free energy principle [18, 19], which formulates cognition and action planning [198] as a process akin to variational inference, similarly involving the minimization of the variational free energy (see next section 2.4.1.1).

2.4.1.1 Evidence Lower Bound

As previously discussed, the Kullback-Leibler divergence defines a special choice in the context of variational inference, as it implies the solution of a variational problem aimed at the minimization of relative entropy and thus information loss. In addition, minimization of the generally intractable Kullback-Leibler divergence is also equivalent to maximizing a lower bound of the log-probability of the data, the so-called *Evidence Lower Bound* (ELBO). If we consider the case where the target distribution is defined by the posterior $p(x|\mathcal{D})$ and substitute $p(x,\mathcal{D})/p(\mathcal{D})$ in the definition of the KL-divergence in Eq. (2.6), we obtain

$$\begin{aligned} D_{KL}[q||p] &= \int \log\left(\frac{q(x)}{p(x|\mathcal{D})}\right) q(x)\,\mathrm{d}x \\ &= \log p(\mathcal{D}) - \underbrace{\mathbb{E}_{q(x)}\left[\log\left(p(\mathcal{D},x)\right) - \log q(x)\right]}_{\mathcal{F}(q)} \end{aligned} \tag{2.33}$$

with $\mathcal{F}$ also being referred to as the negative variational free energy. Due to the non-negativity of the Kullback-Leibler divergence it follows from $\log p(\mathcal{D}) = \mathcal{F}(q) + D_{KL}[q,p]$ that $\mathcal{F}(q) \leq \log p(\mathcal{D})$, with $\mathcal{F}(q) = \mathbb{E}_{q(x)}[\log p(\mathcal{D},x) - \log q(x)]$ the Evidence Lower Bound (alternatively also obtained from Jensen's inequality [199]). While the ELBO generally involves non-tractable expectations which have to be estimated with Monte Carlo, the substitution of the Kullack-Leibler divergence minimization

with the (equivalent) maximization of the ELBO offers the benefit of not depending on the intractable evidence term - we can use the unnormalized distribution $\pi(x)$, which merely shifts the ELBO by a constant. In addition, the estimation of the ELBO $\mathcal{F}$ by means of Monte Carlo provides the ability to monitor convergence (in contrast to, e.g., sampling-based methods). Instead of our initially obtained representation, we can also restructure the ELBO into a different representation containing the entropy term [200].

$$\begin{aligned}\mathcal{F}(q) &= \log p(\mathcal{D}) - D_{KL}[q(x)||p(x|\mathcal{D})] \\ &= \mathbb{E}_{q(x)}[\log p(\mathcal{D}, x)] + \mathbb{H}[q(x)]\end{aligned} \tag{2.34}$$

which suggests that maximization of the Evidence Lower Bound corresponds to identifying a variational approximation $q \in \mathcal{Q}$ which according to the first term Eq. (2.34) *'place[s] high mass on configurations of the latent variables that also explain the observations'* [180], while the second term favors entropic distributions, i.e., which *'maximize uncertainty by spreading their mass on many configurations'* [180].

2.4.1.2 Mean-Field Approximation

One prominent choice regarding the variational approximation $q \in \mathcal{Q}$ lies in the assumption of (partial) independence, defining a restriction of $\mathcal{Q}$ typically due to computational and/or practical consideration. If we introduce a partitioning of x into J disjoint groups $\{x_j, j = 1, ..., J\}$, then the (blocked) mean-field approximation corresponds to the assumption that q factorizes as $q(x) = \prod_{j=1}^{J} q_j(x_j)$ (it defines an approximation because the posterior will generally not possess such a structure). When this mean-field assumption is paired with the Kullback-Leibler divergence one can show that the optimal variational factors $q_j(x_j), j = 1, ..., J$ are implicitly defined by [162, 65]

$$\log q_j^*(x_j) = \mathbb{E}_{i \neq j}[\log p(\mathcal{D}, x)] + \text{const.} \tag{2.35}$$

where $\mathbb{E}_{i \neq j}$ denotes an expectation w.r.t. all the remaining $(J-1)$ variational factors. In certain settings for all (or some) of the variational factors, no further assumptions are necessary, with the family of the variational factors q_j then implicitly defined by Eq. (2.35). In a semi-tractable setting, reintroduction of the optimal factors q^* into the ELBO gives rise to the *collapsed* Evidence Lower Bound (e.g., [201]), which will provide lower-variance estimates and expedited inference.

2.4.1.3 Expectation Maximization

The Expectation-Maximization (EM) [202] is aimed at the identification of point estimates for parameters of intractable probabilistic models and can be derived and mo-

tivated in several ways. Given our preceding discussion in a probabilistic setting, it is most readily introduced by phrasing it as variational inference constrained to specific assumptions regarding mean-field factorization and their variational family. If we consider a partitioning $x = \{y, z\}$. then the ELBO arising for the mean-field approximation $q(y, z) = q(y)\, q(z)$ is given by

$$\log p(\mathcal{D}) = \log \int p(\mathcal{D}|y, z)\, p(y, z)\, \mathrm{d}y\, \mathrm{d}z \tag{2.36}$$

$$\geq \mathbb{E}_{q(y)q(z)} \left[\log \left(\frac{p(\mathcal{D}, y, z)}{q(y)\, q(z)} \right) \right] \tag{2.37}$$

Constraining ourselves to point estimates of y by introducing a Dirac $q(y) = \delta(y - \hat{y})$, then from Eq. (2.35) we find that the optimal variational factor in z is given by the posterior $q^*(z) = p(z|\hat{y}, \mathcal{D})$. The EM-algorithm then corresponds to optimization of the ELBO by iteratively updating the variational factors, i.e., the Dirac and $q^*(z) = p(z|\hat{y}, \mathcal{D})$. Notably, this implies that the traditional EM-algorithm assumes the posterior $p(z|\hat{y}, \mathcal{D})$ to be tractable, as well as generally the required expectations with respect to this distribution. We can summarize the EM-algorithm as iterative optimization requiring an expectation with respect to the posterior as well as a subsequent maximization with respect to $\hat{y}$

E-Step Identify the posterior $q^*(z) = p(z|\hat{y}, \mathcal{D})$ and obtain the lower bound

$$Q(y) = \mathbb{E}_{q^*(z)} \left[\log p(\mathcal{D}, y, z) \right] \tag{2.38}$$

M-Step Maximize the lower bound (or marginal likelihood) to obtain a point estimate

$$\hat{y} = \arg\max_y \mathbb{E}_{q^*(z)} \left[\log p(\mathcal{D}, y, z) \right] \tag{2.39}$$

Here we have neglected any terms of the ELBO which do not depend on y. For applications of the EM-algorithm, z often corresponds to latent variables (e.g. [203]) or to *auxiliary* variables artificially introduced to obtain a tractable expression for the likelihood (e.g. [204]). The Variational-Bayes Expectation-Maximization (VB-EM) algorithm simply defines a relaxation of this algorithm, where instead of being able to identify q^* in the E-step in closed form as the posterior, we conduct variational inference to approximate the posterior; generally, the EM algorithm permits incomplete or sparse updates [205], i.e., none of the two optimization problems implied by the E-step and M-step need to be solved fully in each iteration. The EM algorithm can also be viewed as a coordinate ascent algorithm, iteratively constructing local lower bounds of the log-likelihood in the E-step, and subsequently maximizing this local approximation in the

M-step. Obtaining maximum likelihood point estimates of y in absence of a prior follows from lower-bounding the conditional likelihood $\log p\left(\mathcal{D}|y\right) = \log \int p\left(\mathcal{D}|y,z\right) p\left(z\right) \mathrm{d}z$.

2.4.2 Sampling-based Inference

In contrast to variational inference, the class of sampling-based inference methods is able to obtain a sample-based representation of a target density which in principle can be *asymptotically* accurate. In practice, however, they are often not competitive with variational inference due to their comparably poor scalability as well as the comparably complicated assessment of convergence. The most prominent sampling-based method is given by Markov Chain Monte Carlo (MCMC) [165, 166, 206], which artificially constructs a Markov Chain as an ordered sequence of random variables subject to the Markov property. If carefully constructed to certain criteria, random variables produced by the chain are (correlated) samples from the desired target distribution [207]. The *Metropolis-Hastings algorithm* [208] achieves this by generating proposals from a generally non-symmetric *proposal distribution* $q\left(y|x\right)$, either accepting or rejecting them with probability α

$$x^{(n+1)} = \begin{cases} y^{(n)} & \text{with probability } \alpha\left(x^{(n)}, y^{(n)}\right) \\ x^{(n)} & \text{with probability } 1-\alpha\left(x^{(n)}, y^{(n)}\right) \end{cases} \tag{2.40}$$

with the acceptance probability α given by

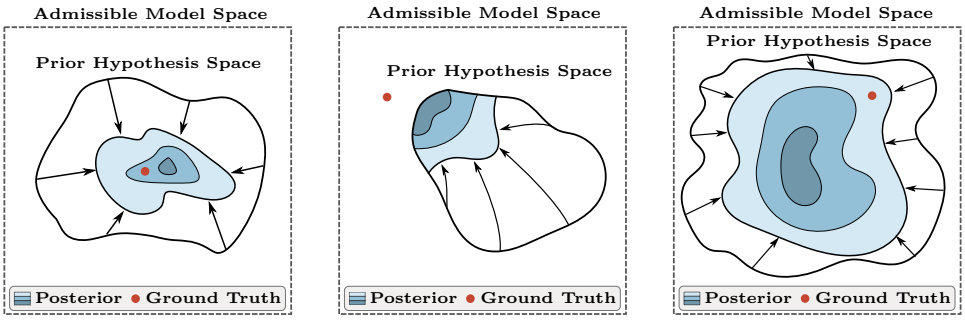
$$\alpha\left(y,x\right) = \min\left\{\frac{p\left(y\right)}{p\left(x\right)}\frac{q\left(x|y\right)}{q\left(y|x\right)}, 1\right\} \tag{2.41}$$

In combination, Eqs. (2.40, 2.41) define the (homogenous) transition operator $t\left(y|x\right)$ of the Markov chain satisfying the *detailed balance* criterium $p\left(y\right)t\left(y|x\right) = p\left(y\right)t\left(x|y\right)$ [165]. As detailed balance is a stronger criterium, this automatically guruantees *invariance* of the target distribution $p\left(x\right)$ under the transition operator, in addition to *irreducability* and *aperiodicity* [165]; more particularly, the target distribution $p\left(x\right)$ is the *only* distribution invariant under the transition operator of the chain (also referred to as stationary distribution). The detailed balance criterium relates to the concept of reversibility and equilibrium thermodynamics (at equilibrium the distribution does not change under time-reversal). As an obvious consequence of sampling proceeding *sequentially* within the Markov process, the samples generated are correlated and as such no longer independent. This implies that any Monte Carlo estimator based on such a correlated sample representation suffers delayed convergence directly related to the extent of the autocorrelation present (quantified by the effective sample size [209],[165]). The convergence of the chain and the quality of the samples produced

as such heavily hinges on the suitability of the proposal distribution $q(y|x)$ given the target distribution; the challenge of obtaining decent sample-representations increases steadily in lockstep with the dimension, the complexity of the submanifold along which most of the probability mass is concentrated, as well as the number and separation of modes of the distribution. In addition, assessment or diagnostic of convergence of Markov chains [210, 211] is generally non-trivial. The acceptance ratio as the fraction of accepted proposals is a coarse metric, with statements regarding optimal values only feasible for a subset of methods under heavily constrained assumptions (e.g., [212]). The dependence of the convergence on the autocorrelation implies that very large or small values of the acceptance ratio will inherently be problematic, and this behavior will be evident in the mixing behavior of the traces of the samples generated by the Markov chain. Many convergence diagnostics such as the Gelman and Rubin Diagnostic [213, 214] rely on analyzing multiple chains, whereas other methods such as e.g. the Geweke diagnostic [215] looks at the temporal evolution of a single chain; generally speaking however assessing and assuring convergence of Markov chains (in finite time) is non-trivial and constitutes a disadvantage compared to variational methods. Apart from discarding the initial samples generated by the chain affected by the choice of the starting point (*burn-in period*), *chain thinning* [216] is another technique that is employed to mitigate sample correlation (it is primarily of utility when memory is of concern, or when computationally intensive operations need to be performed on the generated samples). Research into variants of Metropolis-Hastings has seen many attempts to identify improved proposal distributions yielding small autocorrelation of the chain, while simultaneously being able to deal with other complications (e.g., multimodality of the target distribution). An ensemble of methods such as e.g. Hamiltonian Monte Carlo (HMC) [217] or Metropolis Adjusted Langevin Algorithm (MALA) [218] seek to inject available gradient information into the proposal distribution in order to more efficiently explore and traverse the space (of course this has been extended to the idea of including curvature information as well [219]). Other methodological refinements aim at adaptively tuning the proposal distribution over time, e.g., the adaptive No-U-Turn sampler (NUTS) [220]. Another notable family of methods is defined by *auxiliary* variable methods which apart from HMC also most notably contain Gibbs sampling [221] and slice sampling [222, 223, 224]. Lastly, the inherent sequential nature of Markov Chain Monte Carlo can be elegantly parallelized by making use of Sequential Importance Sampling (SIS), i.e., wrapping the Markov chain within a Sequential Monte Carlo (SMC) method [209, 225]. This is essentially conceptually identical to a particle filter, acting instead on a sequence of artificially defined *bridging distributions* in a static Bayesian inference setting.

2.5 Probabilistic Machine Learning and Deep Architectures

From the Bayesian perspective [226], machine learning simply reduces to probabilistic reasoning for certain types of probabilistic models, and hence any delineation is to some extent artificial. Probabilistic models in the context of machine learning still equate to a hypothesis of the generative process underlying the observed data, and upon observation of the data, we infer the underlying parameters or structure of the model. When we extend this to 'deep learning'', this simply implies an increase in hierarchical complexity of the probabilistic model (e.g. [227, 228]) or underlying neural network architectures [66, 229]). Increasing complexity and expressibility of models by adding depth enjoys advantages compared to adopting shallow but wider models [230], among other reasons due to the ability to encode complex *hierarchical* features. Deep architectures and complex hierarchical features are most commonly constructed by repeatedly stacking elements, ranging from, e.g., relatively simple convolutional layers to more complex 'compound' blocks (e.g. dense blocks [231] or LSTMs [232]). An illustrative example of the complexity of features achievable by deep hierarchical architectures is given by *multi-modal neurons* [233], i.e., neurons that fire and are excitated when confronted with general abstract *concepts*, irrespective of the specific representation (e.g., textual, drawings, or photographs - see for instance the Contrastive Language-Image Pre-Training (CLIP) model [234]). A viewpoint suggested by Tishby [235] relating closely to this observation is that deep learning can be understood as a sequence of operations that iteratively arrives at a more compressed encoding of the pertinent information; considering for instance the example of a classification problem, unnecessary and redundant information is iteratively stripped away until an image is reduced to just a single bit of information (in a binary classification setting). From the previous statement, i.e., probabilistic models defining an a priori hypothesis space regarding the explanation of the observed data, it follows that the specific model assumptions correspond to an *inductive bias*. Consequently, one may argue that specifying models with inductive bias suitable for the task at hand is a large part of successfully applying machine learning methods in challenging problem settings [236]. Exemplarily, the functional mapping identified by a convolutional neural network (CNN) [237] could similarly be expressed by a sufficiently deep and wide dense feedforward neural network. But apart from the apparent and undesirable increase in parameter complexity due to a less parsimonious representation, the feedforward neural network lacks the inductive bias which makes CNNs so useful and successful for certain prediction tasks (e.g. translational invariance [238]). Another way to recognize the hidden inductive bias in the model assumption is the preference of neural networks for simple, low entropy functions [239, 240], as well as the equivalence of many Bayesian neural network architectures with Gaussian Processes, i.e., an a priori belief (and therefore model bias) about function space underlying the mapping [241, 242, 243, 244].



(a) Adequate model (b) Overly simplistic model (c) Overly complex model

Figure 2.5: Figure recreated and adapted from [236], illustrating the impact of model capacity and inductive bias. Subfigures (a,b,c) illustrate the contraction of prior to posterior probability mass in the hypothesis space, with the shift of probability mass indicated by arrows. From left to right: *(a)* The prior hypothesis space envelopes the ground truth without being overly complex, and the inductive bias of the model favors the underlying ground truth. *(b)* An overly simplistic model does not contain the ground truth within the a priori hypothesis space, and consequently the posterior is not able to capture it. *(c)* While an overly complex model with poor inductive bias is (in theory) still able to capture the underlying ground truth, given the limitations of finite data, the posterior mode in the hypothesis space will generally only assign small plausibility to the underlying ground truth.

2.5.1 Deep Learning as Approximate Variational Inference

Building on this discussion, it has been shown that learning with deep neural networks can be readily understood as *'approximate variational inference in a Bayesian setting'* [195]. Similarly, a large ensemble of initially entirely heuristic methods and tools which have contributed considerably to the success and generalization performance of deep learning has subsequently been identified as (highly) approximate Bayesian inference. For instance, stochastic gradient ascent using batch subsampling has been shown to correspond to a stochastic process approximating the posterior [245, 246], with penalty terms for neural network parameters implicitly corresponding to e.g. Gaussian or Laplace priors. Similarly dropout and ensemble methods [247] can be regarded as approximate probabilistic inference, and indeed this observation has given rise to variational inference methods specifically tailored towards the parameters of large and complex neural network architectures [248]. When talking about deep learning from the perspective of probabilistic reasoning, it of course true that most often approximations and simplifying assumptions are introduced compared to a rigorous probabilistic approach (e.g., point estimates or maximum likelihood). Similarly not every conceivable computational method can immediately be subsumed into a probabilistic framework.

Despite this, it can be argued that any learning task is most readily understood from the Bayesian viewpoint (consider, e.g., non-probabilistic support vector machines viewed from the vantage point of relevance vector machines). On the other hand, while learning and inference in deep and complex probabilistic models remain *conceptually* identical to more simplistic settings, there are of course profound complications and challenges arising in practice (e.g., the sheer scale, gradient saturation, non-convex optimizations, learning parameters, non-trivial choices of architectures and hyperparameters, etc.). As a general discussion of probabilistic machine learning methods would be entirely prohibitive in scope, in the following we instead discuss the variational autoencoder as an archetypical example, in particular, due to its similarity to the generative latent variable model adopted by us in section 3.1.

2.5.2 Example: Generative Latent Variable Model

A wide ensemble of generative models [249] with the ability to model complex distributions and interdependencies have been proposed, such as e.g. GANs [250], latent variable models [188, 228, 251, 252], flow-based models [185, 253, 254, 255, 256], diffusion-based models [257], and autoregressive models [258, 259, 260, 261] (with hybrids of course existing, e.g. [262]). These methods come with their own set of advantages and disadvantages, and of course, also differ w.r.t. the technicalities of training and inference (in particular for implicit models [263], which do not permit a closed-form evaluation of the likelihood). One of the desirable features of generative models which we will later exploit is their ability to incorporate unlabeled data and therefore enable semi-supervised learning in a discriminative setting [264, 251, 265]. In the following we briefly discuss the variational autoencoder (VAE) [188] as a prototypical example of generative models. This model posits that each observed datum $x^{(n)}$ within a dataset $\mathcal{D} = \{x^{(n)}\}_{n=1}^{N}$ has been generated by means of sample latent encoding or representation $z^{(n)}$, i.e. a joint density defined by $p(x|z)\,p(z)$. For the simplest case of assumed linear dependence and Gaussianity this gives rise to PPCA [203, 266], and of course with some modifications also permits a semi-supervised variant [267]. If instead of a linear model (as for PPCA) we introduce a nonlinear neural network to parametrize the conditional density $p_\theta(x|z)$ we thereby attain the ability to learn more complex marginal distributions $p_\theta(x)$ at the cost of losing tractability for our model. The variational autoencoder addresses this intractability by means of of an *amortized* variational inference network $q_\Phi(z|x)$ with variational parameters Φ. The amortized auxiliary distribution provides a very scalable [8] way to lower bound the log probability of the data as previously discussed in section (2.4). With the short-hand notation $\mathcal{X} = \{x^{(n)}\}_{n=1}^{N}$ and $\mathcal{Z} = \{z^{(n)}\}_{n=1}^{N}$ we can obtain the evidence lower bound as:

[8] Of course non-amortized or semi-amortized inference [268] could also be adopted

Algorithm 1: SVI for learning a Bayesian Variational Auto-Encoder [188]

while *ELBO not converged* **do**

`// Reparametrization trick`

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

$\mathcal{Z}^{(k)} \leftarrow \varrho_{\Phi}^{\mathcal{Z}}\left(\epsilon_{(k)}\right) \qquad \theta^{(k)} \leftarrow \varrho_{\Phi}^{\theta}\left(\epsilon_{(k)}\right) \qquad k = 1, ..., K$;

`// Monte Carlo estimate of ELBO and its gradient`

Estimate $\mathcal{F} \leftarrow \frac{1}{K}\sum_{k=1}^{K} \hat{\mathcal{F}}\left(\Phi; \mathcal{Z}^{(k)}, \theta^{(k)}\right)$;

$g_{\Phi} \leftarrow \nabla_{\Phi} \frac{1}{K}\sum_{k=1}^{K} \hat{\mathcal{F}}\left(\Phi; \mathcal{Z}^{(k)}, \theta^{(k)}\right)$;

`// Stochastic Gradient Update`

$\Phi^{(n+1)} \leftarrow \Phi^{(n)} + \rho^{(n)} \odot g_{\Phi}$;

$n \leftarrow n + 1$

end

$$\begin{aligned}
\log p_{\theta}(\mathcal{D}) &= \log \int p_{\theta}(\mathcal{D}|\mathcal{Z})\, p(\mathcal{Z})\, \mathrm{d}\mathcal{Z} = \log \int \left(\frac{p_{\theta}(\mathcal{D}, \mathcal{Z})}{q_{\Phi}(\mathcal{Z}|\mathcal{X})}\right) q_{\Phi}(\mathcal{Z}|\mathcal{X})\, \mathrm{d}\mathcal{Z} \\
&\geq \int \log \left(\frac{p_{\theta}(\mathcal{D}, \mathcal{Z})}{q_{\Phi}(\mathcal{Z}|\mathcal{X})}\right) q_{\Phi}(\mathcal{Z}|\mathcal{X})\, \mathrm{d}\mathcal{Z} \qquad \text{(Jensen's inequality)} \\
&= \mathbb{E}_{q_{\Phi}(\mathcal{Z}|\mathcal{X})}\left[\log p_{\theta}(\mathcal{D}|\mathcal{Z})\right] - D_{KL}\left[q_{\Phi}(\mathcal{Z}|\mathcal{X})||p(\mathcal{Z})\right] \\
&:= \mathcal{F}(\theta; \Phi)
\end{aligned} \tag{2.42}$$

The ELBO decouples additively into contributions from individual data points (due to assumed conditional independence). How tightly we lower-bound the log-likelihood is determined by both the approximations gap as well as the amortization gap [269]. The result obtained in Eq. (2.42) suggests simultaneously optimizing the ELBO $\mathcal{F}$ w.r.t. the parameters θ of the generative model as well as the auxiliary variational inference parameters Φ. As this still involves intractable expectations, one resorts to stochastic variational inference and low-variance Monte Carlo estimates of the gradients via the reparametrization trick [187]. Additional noise generally results from also subsampling batches of data due to computational constraints (doubly-stochastic approach). As before in section (2.1.2), instead of a point estimate for θ we can also pursue a fully Bayesian approach by introducing a prior distribution $p(\theta)$ and corresponding variational approximation $q_{\Phi}(\theta)$, with both $q_{\Phi}(\theta)$ and $q_{\Phi}(\mathcal{Z}|\mathcal{X})$ assumed independence and amenable to reparametrization. Noting the assumed independence of the prior, i.e., $p(\mathcal{Z}, \theta) = p(\mathcal{Z})\, p(\theta)$, we again lower bound:

$$\begin{aligned}
\log p(\mathcal{D}) &= \log \int\int p(\mathcal{D}|\mathcal{Z},\theta)\, p(\mathcal{Z})\, p(\theta)\, \mathrm{d}\mathcal{Z}\, \mathrm{d}\theta \\
&\geq \mathbb{E}_{q_\Phi(\mathcal{Z},\theta|\mathcal{X})}\left[\log\left(\frac{p(\mathcal{D}|\mathcal{Z},\theta)\, p(\mathcal{Z})\, p(\theta)}{q_\Phi(\mathcal{Z},\theta|\mathcal{X})}\right)\right] \qquad \text{(Jensen's inequality)} \\
&= \mathbb{E}_{q_\Phi(\mathcal{Z}|\mathcal{X})q_\Phi(\theta)}\left[\log p(\mathcal{D}|\mathcal{Z},\theta)\right] - D_{KL}\left[q_\Phi(\mathcal{Z}|\mathcal{X})\, q_\Phi(\theta)||p(\mathcal{Z})\, p(\theta)\right] \\
&= \mathcal{F}(\Phi)
\end{aligned} \tag{2.43}$$

The intractable expectations in the ELBO are approximated with K Monte Carlo samples (most often $K = 1$), with the KLD term here assumed tractable:

$$\begin{aligned}
\mathcal{F}(\Phi) &\approx \frac{1}{K}\sum_{k=1}^{K}\left[\log p\left(\mathcal{D}|\mathcal{Z}^{(k)},\theta^{(k)}\right)\right] - D_{KL}\left[q_\Phi(\mathcal{Z}|\mathcal{X})\, q_\Phi(\theta)||p(\mathcal{Z})\, p(\theta)\right] \\
&= \frac{1}{K}\sum_{k=1}^{K}\hat{\mathcal{F}}\left(\Phi;\mathcal{Z}^{(k)},\theta^{(k)}\right) \qquad \text{with } \mathcal{Z}^{(k)} \sim q_\Phi(\mathcal{Z}|\mathcal{X}) \;\;,\;\; \theta^{(k)} \sim q_\Phi(\theta)
\end{aligned} \tag{2.44}$$

From the probabilistic viewpoint, the model specifies a joint distribution $p(\mathcal{D},\mathcal{Z},\theta)$ and we reason probabilistically about the plausibility of *unobserved* values of θ and $\mathcal{Z}$ given the dataset $\mathcal{D}$ (see Algorithm 1 for pseudo-code of training the Bayesian variational autoencoder with SVI). Apart from the probabilistic treatment of parameters θ, numerous proposals to extend and modify the variational autoencoder have been made, aiming to extend the method or attempting to improve on limitations and shortcomings (e.g., learning more flexible conditional distribution using normalizing flows [270], or mitigation of uninformative latent code [271]). For the extension of a generative latent variable model towards semi-supervised learning, see for instance [251]. As Generative Adversarial Networks (GANs) [250] became popularized shortly after VAEs and similarly define an archetypical example of generative models, we briefly note that they similarly can be interpreted as latent generators which differ primarily in their method of training. For GANs, the identification of generative model parameters is rephrased as a problem of estimating the ratio of two densities [272] with the help of an auxiliary classifier. Generator and discriminator are then trained jointly until convergence to a Nash equilibrium (see [262] for a hybrid VAE and GAN approach).

2.6 Stochastic Processes and Physical Systems

The goal of this section is to provide a brief outline of the stochastic physical systems which we wish to consider in the context of random media. Given a probability space $(\Omega, \Sigma, \mathbb{P})$ with sample space Ω, σ-Algebra of measurable subsets Σ, and probability measure $\mathbb{P} : \Sigma \to [0, 1]$, we consider a stochastic partial differential equation (SPDE) defined by a general differential operator $\mathcal{L}$ and boundary operator $\mathcal{B}$ dependent on $\omega \in \Omega$

$$\mathcal{L}(s;\omega)\, u = 0 \qquad \forall\ s \in D \tag{2.45}$$

$$\mathcal{B}(s;\omega)\, u = 0 \qquad \forall\ s \in D \tag{2.46}$$

for some d-dimensional computational domain, i.e., $s \in D \subset \mathbb{R}^d$. In Eqs. (2.45 - 2.46), the physical state of the system (e.g., pressure field, velocity field) u consequently also depends on the elementary event $\omega \in \Omega$. As such the scalar, vector or tensor fields defining the solution of the SPDE also constitutes a stochastic process induced by the underlying probability space $(\Omega, \Sigma, \mathbb{P})$. In the context of random media, the dependence of the differential and boundary operator on ω is mediated by a d-dimensional stochastic process describing the random spatial variability of material properties entering into the SPDE. While a d-dimensional stochastic process can generally be seen as a infinite collection of random variables $\{X(t;\omega) : t \in \mathbb{T}\}$ with countable or uncountable d-dimensional index set $\mathbb{T}$ [273], in our specific case the stochastic process takes the form of a random field $G(s;\omega)$ over the computational domain D (i.e., the indexing set $\mathbb{T}$ corresponds to the computational domain D). We are interested in predicting the behavior and propagating uncertainty through the physical system defined by $\mathcal{L}$ and $\mathcal{B}$, or - in a more complex problem setting - to exert a certain degree of control over the system (in the case of the PSP by manipulating the stochastic process, but more generally also by means of additional design variables entering the physical system directly). In the remainder of this section we lay the foundation for our subsequent discussion and numerical investigations, first by introducing the particular case of Gaussian random field $G(s;\omega)$, as well as additionally briefly discussing two specific cases for the differential operator $\mathcal{L}$ corresponding to two different physical phenomena. By doing so we seek to establish the basis for the specific problems and numerical illustrations considered in sections (3.1, 3.2), where we are interested in inferring coarse-grained behavior of physical systems or the effective macroscopic properties of microstructures (assuming that spatial variations of material properties indicated by $G(s;\omega)$ are exhibited on a smaller length-scale). Further details on this *homogenization* or *coarse-graining* process in terms of effective macroscopic properties are given in Appendix A.

2.6.1 Gaussian Random Fields

In the following, we discuss briefly Gaussian random fields as well as a means of sampling them based on their spectral representation (for spatial dimension $d = 2$). A Gaussian random field $G \sim \mathcal{GP}(\mu, \mathcal{C})$ defines a stochastic process where any arbitrary and finite subset of points follows jointly a Gaussian distribution, with the index set $\mathbb{T}$ corresponding to *spatial* coordinates $s \in D$ within the computational domain (hence *field*). In particular, we consider a second order stationary random field characterized by a constant mean function $\mu = \mathbb{E}[G(s,\omega)] = 0$, as well as an autocovariance function depending only on the spatial lag vector $r = s - s'$; this implies that the covariance function can be written as $\mathcal{C}(r) = \mathbb{E}[G(s,\omega)\, G(s+r,\omega)] - \mu^2$. Bochner's theorem [274] asserts that for any continuous and shift-invariant positive kernel the Fourier dual $S(w)$ of the autocovariance function exists, known as the spectral density function (SDF) (where $w \in \mathbb{R}^2$ denotes the wave vector) [275, 276]

$$S(w) = \frac{1}{(2\pi)^2} \int e^{-iw^T r} \mathcal{C}(r)\, \mathrm{d}r \tag{2.47}$$

Under the assumption of a truncation frequency w_{max} (beyond which the SDF diminishes to approximately zero) we can introduce a discrete approximation of the Gaussian process for sampling in the spectral domain [275, 277]

$$\begin{aligned} G(s;\omega) = \sqrt{2} \sum_{l_1=0}^{J_1-1} \sum_{j_2=0}^{J_2-1} \Bigg[& \hat{A}_{j_1,j_2} \cos\left(w_{1,j_1} s_1 + w_{2,j_2} s_2 + \hat{\Psi}_{j_1,j_2}(\omega)\right) \\ & + \tilde{A}_{j_1,j_2} \cos\left(w_{1,j_1} s_1 - w_{2,j_2} s_2 + \tilde{\Psi}_{j_1,j_2}(\omega)\right) \Bigg] \end{aligned} \tag{2.48}$$

with $w_{i,j_i} = j_i \Delta w_i$ and $\Delta w_i = w_{max}/J_i$ (for $i \in \{1,2\}$). The remaining coefficients are defined by:

$$\begin{aligned} \hat{A}_{j_1,j_2} &= \sqrt{C_{j_1,j_2} S(w_{1,j_1}, w_{2,j_2})\, \Delta w_1 \Delta w_2} \\ \tilde{A}_{j_1,j_2} &= \sqrt{C_{j_1,j_2} S(w_{1,j_1}, -w_{2,j_2})\, \Delta w_1 \Delta w_2} \end{aligned} \tag{2.49}$$

$$C_{j_1,j_2} = \begin{cases} \frac{1}{2} & \text{for } j_1 = j_2 = 0 \\ 1 & \text{for } (j_1 = 0 \cap j_2 > 0) \cup (j_1 > 0 \cap j_2) \\ 2 & \text{otherwise} \end{cases} \tag{2.50}$$

Similarly to w_{max} defining the smallest length-scales, the choice of J_1 and J_2 determine Δw_i and consequently the largest length-scale which can be resolved in direction s_i. We refer to to $\hat{\Psi}_{j_1,j_2} \sim \mathcal{U}[0, 2\pi]$ and $\tilde{\Psi}_{j_1,j_2} \sim \mathcal{U}[0, 2\pi]$ as the *phase angles*, and we may summarily denote them by means of a high-dimensional vector $\Psi \in \mathbb{R}^{2J_1J_2}$ in which we stack all phase angles $\hat{\Psi}_{j_1,j_2}$ and $\tilde{\Psi}_{j_1,j_2}$, i.e., $[\Psi]_i \sim \mathcal{U}(0, 2\pi)$ and $i = 1, ..., 2J_1J_2$. If we define a random vector x as corresponding to the values of the random field at a finite number of points in the domain this implies a Gaussian PDF $p(x)$. Assuming furthermore a parametric SDF $S_\varphi(w)$ with process parameters φ induces a conditional PDF $p(x|\varphi)$, accordingly. We can consequently generate samples from this discretized random field representation by sampling uniformly distributed phases angles, and subsequently mapping them as $x = F_\varphi(\Psi)$, where $F_\varphi(\cdot)$ is implicitly defined by Eqs. (2.48) to (2.50). As we model the stochastic process in the spectral domain directly, one particular useful choice which we will adopt in 3.2 is the *spectral mixture kernel*, defining the SDF $S_\varphi(w)$ as a mixture of Gaussians (and φ thus generally given by mixture weights, means and covariances). For sufficiently many Gaussian mixture components in the SDF, any arbitrary stationary covariance function $\mathcal{C}$ can then be approximated to any desired degree of accuracy [278].

Binary random two-phase media

It is a straightforward extension to employ the continuous Gaussian random field to define a stochastic process over binary two-phase random media (see Figure 1.1). While this process-structure linkage is an artificial construction and not directly based on a physical process, it can be used to define a suitable high-dimensional process-structure linkage for evaluating our methodological approach discussed in section 3.2. The definition of the two phases is simply derived from a truncation operation of the $\mathcal{GP}$ w.r.t. a constant x_0, i.e., we define one of the phases as $\mathcal{V}_1(\omega) := \{s \mid G(s, \omega) < x_0\}$ (and $\mathcal{V}_0$ implicitly defined by $\mathcal{V}_1 \cap \mathcal{V}_0 = \emptyset$). Here the constant x_0 implicitly defines the *volume fraction* of the two phases, as - exemplarily for a zero-mean and unit-variance Gaussian Process - in expectation the volume fraction of $\mathcal{V}_1$ is given by $\Phi(x_0)$, with Φ the cumulative density function of the standard Gaussian distribution.

2.6.2 Governing Equations

In the following, we briefly introduce two specific cases for Eq. (2.45) as relevant for our numerical illustrations in sections 3.1 and 3.2. The physical phenomena we consider correspond to either structural phenomena (i.e., deformation under load) as well as diffusion-type processes (e.g., flow through permeable material). In both cases, the PDEs associated with these physical phenomena do of course not permit a closed-form solution and therefore require a suitable numerical discretization. For the work presented in sections 3.1 and 3.2 we adopted the Galerkin approach [279] and made use of Fenics [280, 281] for the discretization and solution of the governing equations

where required. For all cases we consider, we assume that stochasticity enters the partial differential equation by means of material properties described by random fields (*random media*), with the PDE implicitly defining the state variables that satisfy the governing equations for specific realizations of the random field. In consequence, of course, the governing equations outlined in this section relate to the *structure-property* linkage of the PSP chain. The more specific notion of *effective properties* attached to the governing equations as well as their computation by means of Hill's averaging theorem is discussed in Appendix A.

Diffusion We consider as a prototypical case a linear elliptic PDE generally arising for diffusion-type processes such as, e.g., Darcy flow or steady-state heat transfer In this case the abstract differential operator $\mathcal{L}$ from Eq. (2.45) can be written in terms of a Balance Equation (BE) and Constitutive Equation (CE)

$$\text{(BE):} \qquad -\text{div}\,(q) = f \qquad \forall\ s \in D \tag{2.51}$$

$$\text{(CE):} \qquad q = \alpha\,(s;\omega) \cdot \nabla u \qquad \forall\ s \in D \tag{2.52}$$

augmented by suitable Dirichlet and Neumann boundary conditions. The scalar field u describes, e.g., the pressure or temperature over the domain, while the vector field q can be interpreted as (negative) flux. Here f defines a source term over the domain, and the variable coefficient $\alpha\,(s;\omega)$ corresponds to a random field.

Structural Assuming a *linear* constitutive equation, material behavior is defined by

$$\text{(BE):} \qquad -\text{div}\,(\sigma) = f \qquad \forall\ s \in D \tag{2.53}$$

$$\text{(CE):} \qquad \sigma = \mathbb{C}\,(s;\omega) : \epsilon \qquad \forall\ s \in D \tag{2.54}$$

, similarly augmented by suitable Dirichlet and Neumann boundary conditions. The fourth-order stiffness tensor $\mathbb{C}\,(s;\omega)$ in the constitutive equation (CE) relates the tensor-valued stress σ and infinitesimal strain ϵ, which due to its dependence on ω defines a tensor-valued random field. The balance equation (BE) relates the divergence of the stress to a volume force f, i.e., in absence of a force the stress-field needs to be divergence-free over the domain. The stiffness tensor $\mathbb{C}\,(s;\omega)$ corresponds to a tensor-valued random field.

In the next section, we discuss physics-informed machine learning, i.e. incorporation of the constraints articulated by the governing equation into the probabilistic model (as well as their inclusion in an automatic differentiation framework).

2.7 Physics-Informed Learning and Differentiable Physics

When probabilistic reasoning and inference are employed in physical problem settings, the crucial difference lies in the existence of additional *structure* (e.g., expressed by the governing equations as in section 2.6.2) which is absent in purely data-driven problems. Consequently, exploiting this underlying physical structure or ensuring compliance of the model with it may become a major consideration. This may be accomplished by introducing physical features in the loss function directly [72, 282, 283, 284, 285, 286, 287, 288] (e.g. by point-wise enforcement of governing equations for an ensemble of collocation points), or in a Bayesian setting by making the probabilistic model privy to the physical constraints by means of pseudo-observed nodes in the probabilistic graphical model [28, 71, 289]. Similarly, one might construct models which are *inherently* in compliance with certain physical principles such as symmetries or invariances, i.e., they by construction do not permit (or at least mitigiate) violation of certain principles. This can also be seen as a physics-derived or physics-based *inductive bias* of the model (e.g., tensor basis neural networks for turbulence modeling [290, 291], or forcing observations to be explained by a coarse-grained physical model as in section 3.1). By incorporating physical knowledge and/or constraints directly, one can mitigate the dependence on(labeled) data, which generally is expensive to obtain (necessitating either numerical simulations or experimentation). Instead, the information required for inferring the model can be derived directly from the governing equations and underlying physical principles. In combination these methods define the field of *physics-informed machine learning* [73, 292]. Additionally, in a *scientific* or *engineering setting*, one generally faces more stringent requirements regarding both accuracy and reliability of the predictions. Both these criteria strongly suggest the adoption of a probabilistic approach that enables accounting for and quantifying epistemic uncertainty, i.e., providing the ability to assess confidence in predictions as well as specifying confidence bounds. Closely related to the concept of physics-informed machine learning is the concept of *differentiable physics* [293, 294, 295, 296, 297, 298], as it will generally be necessary to backpropagate information through either entire physical simulations (e.g. turbulent flow [299]), or at least to backpropagate through computation of certain physical properties and characteristics of model predictions (in our case exemplarily the flux imbalance in Eq. (2.52)). In both cases, we require the embedding of these physical computations and/or simulations into the probabilistic graphical model and therefore the computational graph defined by it. As such, in the following we set out to provide a cursory discussion of how automatic differentiation as the backbone of probabilistic inference and probabilistic programming can be extended to accommodate physical systems governed by partial differential equations, providing backpropagation of information through the probabilistic model and therefore enabling *gradient-driven* learning and inference.

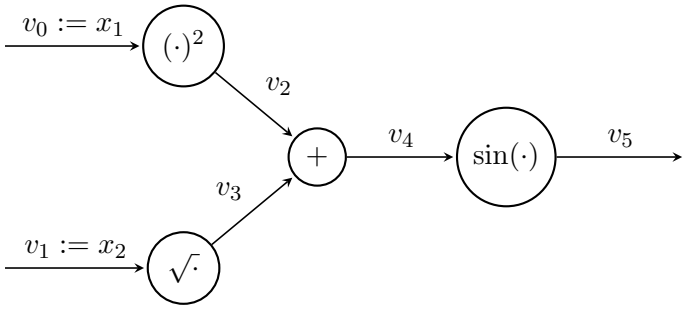


Figure 2.6: The expression $J : \mathbb{R}^n \to \mathbb{R}^m$ given by $J(x_1, x_2) = \sin\left(x_1^2 + \sqrt{x_2}\right)$ with $(n = 2, m = 1)$, decomposed as elementary operations ϕ_j in a directed acyclic graph (DAG), forming the basis for automatic differentiation. The variables $v_j, j = 0, ..., n + p + m - 1$ associated with the edges comprise the $n = 2$ inputs (v_0, v_1) and $m = 1$ outputs (v_5), as well as the p *intermediate* variables resulting from the elementary operations ϕ_j acting on their respective inputs (i.e., here v_2, v_3 and v_4).

2.7.1 Automatic Differentiation and Backpropagation

Automatic differentiation (or also algorithmic differentiation) essentially refers to the automated differentiation of computer programs [300], i.e., providing gradients and higher-order derivatives without any additional implementation overhead and ideally for a numerical cost roughly comparable to the forward pass. In Figure 2.6 we illustrate as a simple example the computation of the expression $\sin\left(x_1^2 + \sqrt{x_2}\right)$. The elementary operations associated with this computation are represented as a *directed acyclic graph* (DAG), which can be seen to form the basis of automatic differentiation. Nodes in this graph correspond to (elemental) operations, while the edges v correspond to inputs and outputs, of preceding and succeeding nodes, respectively. Gradient information can be obtained in more than one way: *forward mode* automatic differentiation is based on the tangent linear code, which may propagate linear perturbations through the model, and thus also directional derivatives. It is typically implemented by overloading the code to not only propagate the actual values v but also simultaneously the perturbations $v^{(1)}$; here $v^{(1)}$ indicates augmented state variables describing sensitivities of any given v. The drawback of this approach is that the numerical cost increases prohibitively for higher dimensions. More useful in our problem setting is *reverse mode* automatic differentiation, which first executes a forward pass to compute all nodal values of the DAG and then uses the linearized model to backpropagate gradient information in the form of the adjoint value $v_{(1)}$ - this is equivalent to the backpropagation algorithm widely employed for neural networks, as well as the adjoint PDE (see discussion in 2.7.2) The reverse mode automatic differentiation using the adjoint involves the forward and backward pass, for which we compute in order

Forward Pass for $j = n, ..., n + p + m - 1$

$$v_j = \phi_j \left(v_i \right)_{i \prec j} \tag{2.55}$$

Backward Pass for every node i in the graph for $i = n + p - 1, ..., 0$

$$v_{(1)i} = \sum_{j : i \prec j} \frac{\partial \phi_j}{\partial v_i} \cdot v_{(1)j} \tag{2.56}$$

Here $i \prec j$ denotes a direct dependence in the graph, while the ϕ_j corresponds to elemental functions (i.e., nodes in the DAG). The ordering of the nodes in the DAG is assumed to reflect the admissible order of computational execution in the DAG. As such Eq. (2.55) simply implies a forward pass for any $j \geq n$ (as the inputs $v_j, j = 0, ..., n-1$ are already known) in order of dependence, with v_j depending on the subset of edges $i \prec j$ which precedes j (dependence indicated by edges). As we retain all values v_j in memory, the backward pass executed in reverse order enables to backpropagate sensitivity information making use of the linearized elementary operations $\partial \phi_j / \partial v_i$, giving rise to the *adjoint* variables $v_{(1)i}$ containing derivatives (or more generally, sensitivities) of the outputs with respect to v_i. Implementation of reverse mode automatic differentiation is most commonly achieved by overloading the forward operations (e.g., [301]), but has also been implemented to operate on compiler-level generated code [302, 303].

2.7.2 Differentiable Physics and the Adjoint Method

Differentiable physics - as it is relevant for our discussion in section 3.1 - involves differentiating the computation of physical features on a fine scale as well as differentiating the solution of a partial differential equation on a coarse scale (using established numerical discretization techniques). We focus on the latter in the following but note that the concept of differentiable physics of course has broader applicability. From the viewpoint of automatic differentiation we can consider the discretized solution of the partial differential equation as a node in the computational graph, mapping inputs $x \in \mathbb{R}^n$ (parameters) to outputs $y \in \mathbb{R}^m$ (discretized representations of e.g. pressure, velocity, etc.). In contrast to explicit layers such as exemplarily feedforward or convolutional layers, the mapping implied by the discretized partial differential equation $y(x) : \mathbb{R}^n \to \mathbb{R}^m$ is only defined *implicitly* by a numerical residual $r(x, y) = 0$ (i.e., effectively an implicit layer [304, 305]). In this context, we do not specify $r(x, y)$ in detail, but simply assume $r \in \mathbb{R}^m$ resulting from a suitable numerical discretization such as the Galerkin method. It is our objective to obtain the gradient of some functional $h(x, y) : \mathbb{R}^{n+m} \to \mathbb{R}$ w.r.t. the parameters x (one could also obtain higher order gradient information, e.g., the action of the Hessian). More generally, we wish to backpropagate gradient information through the PDE solve, if $y(x)$ appears as an

implicit layer within our probabilistic model. If exemplarily the solution of the PDE y defines a flow field depending on the parameters x, backpropagating gradient information through the PDE solver can be understood as the reversal of the flow field, or as the reversal of time itself (for time-dependent problems). As we will see shortly, the adjoint method for partial differential equations is simply equivalent to the reverse mode of automatic differentiation discussed previously. To simplify notation, let us define the *reduced* functional $H(x) = h(x, y(x))$ solely as a function of the parameters x. Then the desired gradient arises as the chain rule applied to the reduced functional

$$\frac{\mathrm{d}H}{\mathrm{d}x} = \frac{\partial h}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}x} + \frac{\partial h}{\partial x} \tag{2.57}$$

The non-trivial term in this equation is given by the derivative of the solution of the PDE with respect to the parameters, i.e. $\mathrm{d}y/\,\mathrm{d}x$. In order to obtain the desired gradient $\mathrm{d}H/\,\mathrm{d}x$ in Eq. (2.57), note that from the requirement for the residual to vanish, i.e., $r(x, y) = 0$, it follows that for any tuples (x, y) satisfying the PDE it follows $\partial r/\partial x = 0$, and therefore

$$\frac{\partial r}{\partial x} = \frac{\partial r}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}x} + \frac{\partial r}{\partial x} = 0 \tag{2.58}$$

From this one obtains directly the tangent linear model (TLM)

$$\frac{\partial r}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}x} = -\frac{\partial r}{\partial x} \tag{2.59}$$

enabling the computation of the Jacobian $J = \mathrm{d}y/\,\mathrm{d}x \in \mathbb{R}^{m\times n}$. The tangent linear equation corresponds to the *forward mode* of automatic differentiation, mapping perturbations δx to δy, thereby enabling the calculation of gradients *irrespective* of the specific functional H. It however unfortunately would imply the repeated solution of a high-dimensional equation system n times (with $x \in \mathbb{R}^n$ generally high-dimensional). The alternative is to employ the adjoint equation - if $\partial r/\partial y$ is invertible we can instead represent the Jacobian as

$$J = \frac{dy}{dx} = -\left(\frac{\partial r}{\partial y}\right)^{-1}\frac{\partial r}{\partial x} \tag{2.60}$$

and substitute this expression for dy/dx into Eq. (2.57)

$$\frac{\mathrm{d}H}{\mathrm{d}x} = \frac{\partial h}{\partial y}\left[-\left(\frac{\partial r}{\partial y}\right)^{-1}\frac{\partial r}{\partial x}\right] + \frac{\partial h}{\partial x} \tag{2.61}$$

Taking the adjoint of this equation and assuming here without loss of generality real-valued vectors and matrices

$$\left(\frac{\mathrm{d}H}{\mathrm{d}x}\right)^T = -\left(\frac{\partial r}{\partial x}\right)^T\left(\frac{\partial r}{\partial y}\right)^{-T}\left(\frac{\partial h}{\partial y}\right)^T + \left(\frac{\partial h}{\partial x}\right)^T \tag{2.62}$$

We can define the *adjoint* vector $\lambda = \frac{\partial r}{\partial y}^{-T}\frac{\partial h}{\partial y}^T \in \mathbb{R}^m$, as defined by:

$$\left(\frac{\partial r}{\partial y}\right)^T \lambda = \frac{\partial h}{\partial y}^T \tag{2.63}$$

This constitutes the (linear) adjoint equation. Once we have solved this equation for λ we can rephrase Eq. (2.62) w.r.t. λ

$$\left(\frac{\mathrm{d}H}{\mathrm{d}x}\right)^T = -\frac{\partial r}{\partial x}^T \lambda + \left(\frac{\partial h}{\partial x}\right)^T \tag{2.64}$$

Solving the partial differential equation and subsequently solving the adjoint equation to obtain the derivative $\mathrm{d}H/\,\mathrm{d}x$ corresponds to the *backward mode* of automatic differentiation. After the forward pass and during the backward pass of the computational graph, all the pertinent quantities x, y and $\partial h/\partial y$ appearing as a source term in the adjoint equation are known. Hence gradients can be backpropagated, with the cost being dominated by the solution of the *linear* adjoint equation (in addition to the forward pass). For any higher-dimensional setting of $x \in \mathbb{R}^n$, the adjoint equation corresponding to the backward mode automatic differentiation is generally decidedly preferable in terms of numerical cost.

3
Summary of Publications

This chapter comprises short summaries of the following two publications which are the subject of this cumulative dissertation:

Paper A: Rixner, Maximilian, and Phaedon-Stelios Koutsourelakis. "A probabilistic generative model for semi-supervised training of coarse-grained surrogates and enforcing physical constraints through virtual observables." *Journal of Computational Physics* 434 (2021): 110218.

Paper B: Rixner, Maximilian, and Phaedon-Stelios Koutsourelakis. "Self-supervised optimization of random material microstructures in the small-data regime." *npj Computational Materials* 8.1 (2022): 1-11.

The publications are attached in full in Appendix B and C.

3.1 Paper A

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

M. Rixner, P.S. Koutsourelakis

Summary

Computational physics is concerned with modeling complex physical phenomena and processes, in the most pertinent cases spanning a wide range of spatio-temporal scales. The numerical resolution of such a hierarchy of scales in fine-grained models is however also inextricably tied to a significant computational burden. While this suggests the construction of cheap surrogates, identifying a suitable surrogate in high-dimensional settings with limited availability of labeled data in itself poses a grand challenge. In this paper, we suggest a novel approach for surrogate construction in the context of random media. The generative model underlying our proposed surrogate is endowed with a physics-based inductive bias, as predictions are forced to be informed by coarse-grained, effective physical properties (which in turn are derived from a compressed latent representation of the random media). The information bottleneck induced by this architecture is moreover not merely informed by labeled data, but additionally incorporates unlabeled data for semi-supervised learning as well as physical constraints and/or inequalities in the form of virtual observables. The bottlenecked architecture as such forces the dense and physics-constrained accumulation of information from all the different sources available (i.e., labeled data, unlabeled data, governing equations). Notably, both the coarse-grained physical model as well as the virtual observables encoding the governing equations are embedded directly within the probabilistic graphical model, with the coarse-grained physic model inferred by backpropagating in a differentiable-physics setting.

Contributions

(MR): conceptualization, physics and machine-learning modeling and computations, algorithmic and code development, writing of the paper. **(PSK)**: conceptualization, writing of the paper.

Reference

Rixner, Maximilian, and Phaedon-Stelios Koutsourelakis. "A probabilistic generative model for semi-supervised training of coarse-grained surrogates and enforcing physical constraints through virtual observables." *Journal of Computational Physics* 434 (2021): 110218.

3.2 Paper B

Self-supervised optimization of random material microstructures in the small-data regime.

M. Rixner, P.S. Koutsourelakis

Summary

The stochastic inversion of the process-structure-property (PSP) chain corresponds to the identification of process parameters yielding optimal effective material properties on a macroscopic scale. It defines a largely unsolved problem and an object of ongoing research, as any attempt to approach it is confronted with both its inherent intractability as well as the complexities of the physical linkages. These linkages - mapping process to structure and structure to property - are generally non-deterministic and expensive to resolve numerically. This paper suggests an approach in which a convolutional neural network acts as a discriminative surrogate, wrapped within an optimization algorithm iteratively lower-bounding the intractable objective function. Lower bounding is achieved by means of stochastic variational inference, simultaneously providing noisy but comparably low-variance Monte Carlo estimates of the gradient of the objective w.r.t. process parameters (even in a high-dimensional setting). To mitigate the dependence on expensive labeled training data (requiring numerical simulations), the dataset informing the surrogate is successively enriched using an active learning approach embedded within the optimization routine. The addition of new data points is guided and informed by an acquisition function that is coupled to the objective function itself, i.e., the informativeness of microstructure-property pairs is tied directly to the optimization problem at hand. Basing our numerical illustrations on the spectral representation of a thresholded Gaussian Process for the process-structure map (i.e., binary-two phase microstructures), we execute and evaluate this approach for different flexible notions of optimality, solving a material's design problem by stochastic inversion of the full PSP chain for thermal and structural properties in a high-dimensional setting.

Contributions

(MR): conceptualization, physics and machine-learning modeling and computations, algorithmic and code development, writing of the paper. **(PSK)**: conceptualization, writing of the paper.

Reference

Rixner, Maximilian, and Phaedon-Stelios Koutsourelakis. "Self-supervised optimization of random material microstructures in the small-data regime." *npj Computational Materials* 8.1 (2022): 1-11.

4

Discussion and Outlook

In the following, we provide a brief concluding discussion of the probabilistic models which were developed in this thesis for the prediction and control of materials systems in the context of random media. We emphasize that the methods under investigation correspond to the emerging approach of *learning* or *infering* the behavior of physical systems [68, 69, 70], instead of relying on numerical discretization techniques or hand-crafted models defined by experts. In our context, this entails the identification of a parsimonious set of features predictive of the coarse-grained behavior of the physical system. The ability to predict the behavior of a system with a comparably cheap probabilistic surrogate subsequently enables to drive the solution of many-query applications, such as optimization problems or inverse problems. In all our investigations, it was of primary concern to mitigate the number of datapoints required for the construction of our surrogate model. This concern for a data-parsimonious approach is represented by the utilization of techniques such as active learning, semi-supervised learning, and physics-informed learning. In such a setting one either seeks to maximize the information contained within a fixed number of data points (active learning), derives additional information from unlabeled data (semi-supervised learning), or injects a priori knowledge from governing equations directly (physics-informed learning). The reason for this emphasis on data reduction derives from the fact that computational physics resides within the small-data regime, i.e., apart from data generally being high-dimensional it also needs to be considered scarce due to the expense of obtaining it (by means of numerical simulations).

In section 3.1 we demonstrated the ability to predict the coarse-grained response of a parametric partial differential equation in the context of random media (structure-property linkage), thereby enabling the propagation of uncertainty in a high-dimensional setting with extremely few labeled datapoints. In order to obtain meaningful generalization performance in such a challenging setting, the probabilistic model was forced to make predictions by identifying effective properties of a physical coarse-grained model, instead of purely relying on black-box models (this can be regarded as an information bottleneck as well as a strong physics-derived inductive bias, see Figure 2.5). To this end, the physical coarse-grained model and its adjoint were embedded directly in the convolutional encoder-decoder architecture of the generative model, i.e., effective properties were derived from the latent representation of the information bottleneck. In

addition to the incorporation of the coarse-grained model, the other pivotal feature of our approach was defined by the adoption of a generative model which allowed the *simulatenous* incorporation of labeled data, unlabeled data, as well as physical constraints derived from governing equations. Information from these different sources was accumulated and fused in our information bottleneck, leading to their various respective contributions to the Evidence Lower Bound (resulting from a stochastic variational inference approach for training the model). Importantly, we also demonstrated the increase of predictive performance by incorporating only a *reduced* set of coarse-grained physical constraints. While the incorporation of the governing equation at full resolution into the machine learning model provides in principle the maximum of information, it also implies the largest numerical cost - thereby counteracting one of the foremost reasons for physics-informed learning (the expense of obtaining labeled data). In contrast, we were able to achieve significant improvement in predictive performance while only considering a projection of the residual, corresponding to a certain filtered view of the governing equations (in our case translating to numerically efficient low-rank updates of the variational mean-field updates).

In section (3.2) the prediction of the coarse-grained response of the materials system was enveloped within a more complex optimization objective, demonstrating the identification of optimal process parameters according to desirable effective material properties. I.e., we addressed the challenging task defined by the stochastic inversion of the *complete* process-structure-property chain. Our proposed formulation was able to incorporate a diverse ensemble of optimization objectives (maximizing an expected utility, or minimizing the Kullback-Leibler divergence for a specified target distribution of the effective, macroscopic properties). Intractability, numerical cost and non-differentiability of the optimization objective were addressed by the simultaneous introduction of a probabilistic discriminative convolutional surrogate, as well as a stochastic variational inference approach iteratively lower-bounding the intractable objective (Variational-Bayes Expectation-Maximization algorithm). While here the surrogate was purely data-driven and not privy to physical information, the VB-EM algorithm was embedded within an outer-loop data acquisition step informed by an optimization-specific acquisition function (thereby enabling selection of the most informative microstructure-property pairs for an active learning approach). Apart from mitigating the numerical cost of generating training data, this incremental enrichment of the dataset was also rooted in the general futility of constructing a globally valid surrogate in a setting where a priori unknown process parameters will heavily impact the nature of the generated microstructures. Instead, we suggested expanding the predictive utility of the surrogate according to necessity as we traverse the process parameter space during optimization. While this approach cumulatively required several thousand data points for the stochastic inversion of the PSP chain, this still constitutes comparably few datapoints when considering the complexity of the linkages as well as the high-dimensional nature of the microstructures (i.e., small-data domain). We

particularly emphasize the high-dimensionsal nature of the process parameters in our numerical demonstrations, which significantly exceeded prior published work [29]. In conclusion, both methodological frameworks developed in sections 3.1 and 3.2, may be regarded as probabilistic models resolving the complete or partial process-structure-property chain for the purpose of prediction and/or control. Having summarized some of the novel characteristics and contributions of our proposed methods, we conclude by discussing some of the insights obtained during their investigation, as well as correspondingly some modifications or further developments which could be attempted:

Spatial Awareness The methods under investigation made use of convolutional neural networks for extracting pertinent features from the discretized representations of the random media, either in a discriminative or generative setting. While architectures based on convolutional features inherently offer a certain level of spatial coherence, improvements could be made by increasing spatial awareness as well as ideally expanding the ability to identify multi-scale features. Classes of models which are inherently suitable in this regard are, e.g., given by autoregressive methods [259, 258] as well as Gaussian Process [33, 306] based approaches. As an example, the discriminative surrogate in section 3.2 was restricted to capturing dependencies limited by the depth and filter size of the convolutional architecture (see for instance dilated convolutions [307] for increased receptive fields). Similarly in 3.1 the nature of the architecture was not inherently aware of the effective properties defining a spatial field, and instead just regarded it as an unstructured vector derived from the latent representation. Instead, one might e.g. use a Gaussian Process parametrized by inducing points [201] to define a mapping between latent space and effective coarse-grained properties that has spatial awareness of the effective properties.

Morphological Awareness With regards to morphological awareness, we note that in the context of generative modeling it is not necessarily desirable to identify a compressed encoding or representation of the random media which enables a pixel-perfect reconstruction (as implied by our likelihood in 3.1). An alternative would be to incentivize the model to retain characteristical statistical properties, particularly in the context of random heterogeneous media with distinct phases (see e.g. reduced order models based on 2-point statistics [23]). While any semi-supervised learning approach will bias latent representations towards disregarding irrelevant fine-scale variations in favor of predictive features, it seems plausible to the author that a reconstruction likelihood aware of the morphological equivalence [308, 309, 310, 311] could yield further improvements, e.g., by scoring microstructures based on the Gram matrix of convolutional features similar to neural style transfer [312, 313]. In such a setting the latent representation will be incentivized to encode complex statistical features rather than memorize pixel-

perfect representations. We note that conceptually similar challenges appear in the setting of learning and synthesizing textures (e.g. [314, 315]).

Representation The pixelized discrete representation of microstructures and random media which we adopted is not definitive but simply just constitutes one possible modeling choice. While certainly the most common approach [316, 317, 318, 319], depending on the microstructure under consideration it might prove advantageous to consider alternatives representations. Exemplarily, for polycrystalline microstructures, one could adopt a sparse graph representation encoding neighbourship relations of grains as well as grain features (such as orientation and size as node-level information). The advantage would be a more natural and parsimonious description, due to the close equivalence of graphs resulting from microstructures with similar *statistical* properties (see the previous discussion of morphological awareness). In addition, and maybe more importantly, it will lead to a significantly lower-dimensional representation of the microstructures. A graph-based representation of microstructures would then suggest the adoption of graph neural networks [320] for the probabilistic model.

Probabilistic Models and Inference A further avenue of improvement would be given by more flexible and expressive variational inference approaches, which were employed either in the training of the generative model or the stochastic inversion of the PSP chain (e.g. normalizing flows [185, 321]). In the context of learning a probabilistic model for the process-structure-property chain, it also seems promising to investigate inherently invertible architectures such as flow-based models [253, 254, 255, 256] (invertibility defines a desirable property in this setting where the ultimate goal is defined by the inversion of the model). Flow-based models can also incorporate autoregression [260] and multiscale features [261], which as previously discussed are attractive in the context of random media. Regarding physics-informed machine learning in the context of probabilistic models and inference, we mention that any model is advantageous which enables to concentrate a priori probability mass on a relatively low-dimensional manifold (see e.g. impact of the chosen prior distribution for Bayesian Conjugate Gradient [9]). This would for instance translate to the adoption of a low-rank Gaussian for the noise model in section 3.1.

Bayesian Neural Networks In our approach, neural network parameters have been relegated to point estimates (both for discriminative as well as generative models). A fully Bayesian approach [241] seeking a variational approximation (e.g. [322, 186]) should yield further generalization improvements in the small data domain, as well as provide the ability to more tightly and more consistently capture predictive uncertainty (in particular it would enable to provide uncertainty bounds for the surrogate-based objective for the PSP inversion). Additionally, the

Bayesian treatment of neural network parameters also yields a more information-rich environment for adaptive acquisition of data points by opening the door to Bayesian active learning [323], i.e., enabling the introduction of acquisition functions differentiating between inherent noise and epistemic uncertainty of neural network parameters (see for instance 'Bayesian Active Learning by Disagreement' [324, 325]). Bayesian Active Learning also relates to Bayesian Experimental Design and information-theoretic considerations introduced in section (2.2.2); one may for instance seek to introduce new datapoints which exhibit high mutual information with the neural network parameters, i.e., offer the highest expected information gain [326].

For the stochastic inversion of the process-structure-property chain (section 3.1), we also mention the possibility of a more integrated approach by constructing a generative model encompassing process parameters, microstructures and effective properties. Among some other benefits, it would enable tighter coupling compared to the current inner-loop outer-loop approach enveloping a discriminative model (i.e., data acquisition loop and EM loop). If such a unified generative model is attainable, it would be a more natural choice to additionally also infer the process-structure map (as opposed to daisy chaining two discriminative models). Closer to the confines of the currently adopted choices, a more incremental expansion enabling more complex problem setting could be given by the adoption of a mixture of experts [327, 328, 329] for the discriminative surrogate. In the context of the generative model and physics-informed machine approach in section 3.2, we observe that the physical information infused into the generative model by means of the virtual observables was fixed a priori. An *adaptive* selection of virtual observables based on information-theoretic consideration or adversarial examples [330] could enable us to select the most informative queries given a fixed computational budget. A more drastic (yet interesting) modification would be to treat the differential operator itself as an unknown entity with the possibility of constructing a suitable prior [331, 332]. In closing we remark that more flexible physics-informed approaches could be adopted by turning toward compiler-level automatic differentiation [333, 302, 303], as the differentiable coupling of physical computations and probabilistic model poses non-trivial practical constraints. In the opinion of the author, physics-informed probabilistic methods in the context of partial differential equations also enjoys several connections to, e.g., probabilistic numerics [8], randomized/probabilistic linear algebra [334, 335] as well as probabilistic solvers [11, 10], which remain to be explored in this setting.

Bibliography

[1] Edwin T Jaynes. *Probability theory: The logic of science*. Cambridge university press, 2003.

[2] David Barber. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

[3] R.C. Smith. *Uncertainty Quantification: Theory, Implementation, and Applications*. Computational Science and Engineering. SIAM, 2013.

[4] T.J. Sullivan. *Introduction to Uncertainty Quantification*. Texts in Applied Mathematics. Springer International Publishing, 2015.

[5] Kirthevasan Kandasamy, Jeff Schneider, and Barnabás Póczos. Query Efficient Posterior Estimation in Scientific Experiments via Bayesian Active Learning. *Artif. Intell.*, 243(C):45–56, February 2017.

[6] Alexis Bondu, Vincent Lemaire, and Marc Boullé. Exploration vs. exploitation in active learning: A bayesian approach. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2010.

[7] Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical Science*, pages 273–304, 1995.

[8] Philipp Hennig, Michael A Osborne, and Mark Girolami. Probabilistic numerics and uncertainty in computations. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471(2179):20150142, 2015.

[9] Jon Cockayne, Chris Oates, Ilse Ipsen, and Mark Girolami. A bayesian conjugate gradient method, 2018.

[10] Simon Bartels, Jon Cockayne, Ilse CF Ipsen, Mark Girolami, and Philipp Hennig. Probabilistic linear solvers: A unifying view. *arXiv preprint arXiv:1810.03398*, 2018.

[11] Jonathan Wenger and Philipp Hennig. Probabilistic linear solvers for machine learning. *arXiv preprint arXiv:2010.09691*, 2020.

[12] Michael Schober, David K Duvenaud, and Philipp Hennig. Probabilistic ode solvers with runge-kutta means. *Advances in neural information processing systems*, 27, 2014.

[13] Sebastian Brandstaeter, Sebastian L Fuchs, Roland C Aydin, and Christian J Cyron. Mechanics of the stomach: A review of an emerging field of biomechanics. *GAMM-Mitteilungen*, 42(3):e201900001, 2019.

[14] Sebastian Brandstaeter, Alessio Gizzi, Sebastian L Fuchs, Amadeus M Gebauer, Roland C Aydin, and Christian J Cyron. Computational model of gastric motility with active-strain electromechanics. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift Für Angewandte Mathematik Und Mechanik*, 98(12):2177–2197, 2018.

[15] Julia M Hoermann, Cristóbal Bertoglio, Martin Kronbichler, Martin R Pfaller, Radomir Chabiniok, and Wolfgang A Wall. An adaptive hybridizable discontinuous galerkin approach for cardiac electrophysiology. *International journal for numerical methods in biomedical engineering*, 34(5):e2959, 2018.

[16] Stephen B Pope and Stephen B Pope. *Turbulent flows*. Cambridge university press, 2000.

[17] Claude E Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, 2001.

[18] Karl Friston. The free-energy principle: a unified brain theory? *Nature reviews neuroscience*, 11(2):127–138, 2010.

[19] Sebastian Gottwald and Daniel A Braun. The two kinds of free energy and the bayesian revolution. *PLoS computational biology*, 16(12):e1008420, 2020.

[20] P Suquet, QS Nguyen, and P Germain. Continuum thermodynamics. *Trans. ASME. J. Appl. Mech. 50, 1010*, 1020, 1983.

[21] Chris Preston. *Random fields*, volume 534. Springer, 2006.

[22] Phadeon-Stelios Koutsourelakis. Probabilistic characterization and simulation of multi-phase random media. *Probabilistic Engineering Mechanics*, 21(3):227–234, 2006.

[23] Noah H Paulson, Matthew W Priddy, David L McDowell, and Surya R Kalidindi. Reduced-order structure-property linkages for polycrystalline microstructures based on 2-point statistics. *Acta Materialia*, 129:428–438, 2017.

[24] Salvatore Torquato and HW Haslach Jr. Random heterogeneous materials: microstructure and macroscopic properties. *Appl. Mech. Rev.*, 55(4):B62–B63, 2002.

[25] Helmut Clemens, Svea Mayer, and Christina Scheu. Microstructure and properties of engineering materials. *Neutrons and synchrotron radiation in engineering materials science: From fundamentals to applications*, pages 1–20, 2017.

[26] Constantin. Grigo and Phaedon-Stelios. Koutsourelakis. Bayesian Model and Dimension Reduction for Uncertainty Propagation: Applications in Random Media. *SIAM/ASA Journal on Uncertainty Quantification*, 7(1):292–323, January 2019.

[27] Constantin Grigo and Phaedon-Stelios Koutsourelakis. A physics-aware, probabilistic machine learning framework for coarse-graining high-dimensional systems in the Small Data regime. *Journal of Computational Physics*, 397:108842, November 2019.

[28] Sebastian Kaltenbach and Phaedon-Stelios Koutsourelakis. Incorporating physical constraints in a deep probabilistic machine learning framework for coarse-graining dynamical systems. *Journal of Computational Physics*, 419:109673, October 2020.

[29] Anh Tran and Tim Wildey. Solving stochastic inverse problems for property–structure linkages using data-consistent inversion and machine learning. *JOM*, 73(1):72–89, 2021.

[30] Anh Tran, John A Mitchell, Laura P Swiler, and Tim Wildey. An active learning high-throughput microstructure calibration framework for solving inverse structure–process problems in materials informatics. *Acta Materialia*, 194:80–92, 2020.

[31] Maximilian Rixner and Phaedon-Stelios Koutsourelakis. Self-supervised optimization of random material microstructures in the small-data regime. *npj Computational Materials*, 8(1):1–11, 2022.

[32] A O'Hagan and MC Kennedy. Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13, 03 2000.

[33] CE. Rasmussen and CKI. Williams. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, USA, January 2006.

[34] Ilias Bilionis, Nicholas Zabaras, Bledar A. Konomi, and Guang Lin. Multi-output separable Gaussian process: Towards an efficient, fully Bayesian paradigm for uncertainty quantification. *Journal of Computational Physics*, 241:212 – 239, 2013.

[35] D. Xiu and G. Karniadakis. The Wiener–Askey Polynomial Chaos for Stochastic Differential Equations. *SIAM Journal on Scientific Computing*, 24(2):619–644, 2002.

[36] Jin Meng and Heng Li. Efficient uncertainty quantification for unconfined flow in heterogeneous media with the sparse polynomial chaos expansion. *Transport in Porous Media*, 126(1):23–38, 2019.

[37] Youssef M Marzouk and Habib N Najm. Dimensionality reduction and polynomial chaos acceleration of bayesian inference in inverse problems. *Journal of Computational Physics*, 228(6):1862–1902, 2009.

[38] Justin Gregory Winokur. *Adaptive sparse grid approaches to polynomial chaos expansions for uncertainty quantification*. PhD thesis, Duke University, 2015.

[39] Michael Eldred. Recent advances in non-intrusive polynomial chaos and stochastic collocation methods for uncertainty analysis and design. In *50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference 17th AIAA/ASME/AHS Adaptive Structures Conference 11th AIAA No*, page 2274, 2009.

[40] Géraud Blatman and Bruno Sudret. Adaptive sparse polynomial chaos expansion based on least angle regression. *Journal of computational Physics*, 230(6):2345–2367, 2011.

[41] Peter Benner, Serkan Gugercin, and Karen Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM review*, 57(4):483–531, 2015.

[42] Dominic Soldner, Benjamin Brands, Reza Zabihyan, Paul Steinmann, and Julia Mergheim. A numerical study of different projection-based model reduction techniques applied to computational homogenisation. *Computational mechanics*, 60(4):613–625, 2017.

[43] Alfio Quarteroni, Andrea Manzoni, and Federico Negri. *Reduced basis methods for partial differential equations: an introduction*, volume 92. Springer, 2015.

[44] Jan S Hesthaven, Gianluigi Rozza, Benjamin Stamm, et al. *Certified reduced basis methods for parametrized partial differential equations*, volume 590. Springer, 2016.

[45] Eric J Parish, Christopher R Wentland, and Karthik Duraisamy. The adjoint petrov–galerkin method for non-linear model reduction. *Computer Methods in Applied Mechanics and Engineering*, 365:112991, 2020.

[46] Kevin Carlberg, Charbel Bou-Mosleh, and Charbel Farhat. Efficient non-linear model reduction via a least-squares petrov–galerkin projection and compressive tensor approximations. *International Journal for Numerical Methods in Engineering*, 86(2):155–181, 2011.

[47] David Galbally, Krzysztof Fidkowski, Karen Willcox, and Omar Ghattas. Non-linear model reduction for uncertainty quantification in large-scale inverse problems. *International journal for numerical methods in engineering*, 81(12):1581–1608, 2010.

[48] T. Cui, Y. Marzouk, and K. Willcox. Data-driven model reduction for the bayesian solution of inverse problems. *International Journal for Numerical Methods in Engineering*, 102(5):966–990, 2015.

[49] Mengwu Guo and Jan S Hesthaven. Reduced order modeling for nonlinear structural analysis using gaussian process regression. *Computer methods in applied mechanics and engineering*, 341:807–826, 2018.

[50] Martin A Grepl, Yvon Maday, Ngoc C Nguyen, and Anthony T Patera. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 41(3):575–605, 2007.

[51] Tiangang Cui, James Martin, Youssef M Marzouk, Antti Solonen, and Alessio Spantini. Likelihood-informed dimension reduction for nonlinear inverse problems. *Inverse Problems*, 30(11):114015, 2014.

[52] Tiangang Cui, Kody JH Law, and Youssef M Marzouk. Dimension-independent likelihood-informed mcmc. *Journal of Computational Physics*, 304:109–137, 2016.

[53] Tan Bui-Thanh and Mark Girolami. Solving large-scale pde-constrained bayesian inverse problems with riemann manifold hamiltonian monte carlo. *Inverse Problems*, 30(11):114014, 2014.

[54] Clarence W. Rowley, Tim Colonius, and Richard M. Murray. Model reduction for compressible flows using POD and Galerkin projection. *Physica D: Nonlinear Phenomena*, 189(1):115 – 129, 2004.

[55] Kookjin Lee and Kevin T Carlberg. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *Journal of Computational Physics*, 404:108973, 2020.

[56] J Nagoor Kani and Ahmed H Elsheikh. Dr-rnn: A deep residual recurrent neural network for model reduction. *arXiv preprint arXiv:1709.00939*, 2017.

[57] Min Wang, Siu Wun Cheung, Wing Tat Leung, Eric T Chung, Yalchin Efendiev, and Mary Wheeler. Reduced-order deep learning for flow dynamics. the interplay between deep learning and model reduction. *Journal of Computational Physics*, 401:108939, 2020.

[58] J. Hesthaven and S. Ubbiali. Non-intrusive reduced order modeling of nonlinear problems using neural networks. *Journal of Computational Physics*, 363:55 – 78, 2018.

[59] Michael B Giles. Multilevel monte carlo methods. *Acta Numerica*, 24:259–328, 2015.

[60] P.-S. Koutsourelakis. Accurate Uncertainty Quantification Using Inaccurate Computational Models. *SIAM Journal on Scientific Computing*, 31(5):3274–3300, 2009.

[61] P. Perdikaris, D. Venturi, J. O. Royset, and G. E. Karniadakis. Multi-fidelity modelling via recursive co-kriging and Gaussian–Markov random fields. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471(2179):20150018, 2015.

[62] Jonas Nitzler, Jonas Biehler, Niklas Fehn, Phaedon-Stelios Koutsourelakis, and Wolfgang A. Wall. A generalized probabilistic learning approach for multi-fidelity uncertainty propagation in complex physical simulations, 2020.

[63] Jonas Biehler, Michael W Gee, and Wolfgang A Wall. Towards efficient uncertainty quantification in complex and large-scale biomechanical problems based on a bayesian multi-fidelity scheme. *Biomechanics and modeling in mechanobiology*, 14(3):489–513, 2015.

[64] Christoph Striegel, Jonas Biehler, Wolfgang A Wall, and Göran Kauermann. A multifidelity function-on-function model applied to an abdominal aortic aneurysm. *Technometrics*, pages 1–12, 2022.

[65] C.M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, 2006.

[66] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[67] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[68] Steven L Brunton and J Nathan Kutz. *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2019.

[69] Michael Frank, Dimitris Drikakis, and Vassilis Charissis. Machine-Learning Methods for Computational Science and Engineering. *Computation*, 8(1):15, March 2020.

[70] P. S. Koutsourelakis, N. Zabaras, and M. Girolami. Special Issue: Big data and predictive computational modeling. *Journal of Computational Physics*, 321:1252–1254, September 2016.

[71] Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

[72] Y. Zhu, N. Zabaras, P.-S. Koutsourelakis, and P. Perdikaris. Physics-Constrained Deep Learning for High-dimensional Surrogate Modeling and Uncertainty Quantification without Labeled Data. *Journal of Computational Physics*, 2019. submitted for publication.

[73] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

[74] C. Robert. *The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation*. Springer Texts in Statistics. Springer New York, 2007.

[75] David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[76] Sheldon M Ross. *Introduction to probability models*. Academic press, 2014.

[77] Jay L Devore. *Probability and Statistics for Engineering and the Sciences*. Cengage learning, 2011.

[78] Bruno A Olshausen. Bayesian probability theory. *The Redwood Center for Theoretical Neuroscience, Helen Wills Neuroscience Institute at the University of California at Berkeley, Berkeley, CA*, 2004.

[79] Zoubin Ghahramani. Probabilistic machine learning and artificial intelligence. *Nature*, 521(7553):452–459, 2015.

[80] K.P. Murphy. *Machine Learning: A Probabilistic Perspective*. Adaptive computation and machine learning. MIT Press, 2012.

[81] Richard T Cox. Probability, frequency and reasonable expectation. *American journal of physics*, 14(1):1–13, 1946.

[82] A.D. Aleksandrov, A.N. Kolmogorov, and M.A. Lavrent'ev. *Mathematics: Its Content, Methods and Meaning*. Dover Books on Mathematics. Dover Publications, 2012.

[83] Stefan Arnborg and Gunnar Sjödin. On the foundations of bayesianism. In *AIP Conference Proceedings*, volume 568, pages 61–71. AIP, 2001.

[84] Roberto Trotta. Bayes in the sky: Bayesian inference and model selection in cosmology. *Contemporary Physics*, 49(2):71–104, 2008.

[85] William Briggs. *Uncertainty: the soul of modeling, probability & statistics*. Springer, 2016.

[86] Zichao Long, Yiping Lu, Xianzhong Ma, and Bin Dong. Pde-net: Learning pdes from data. *arXiv preprint arXiv:1710.09668*, 2017.

[87] Zichao Long, Yiping Lu, and Bin Dong. Pde-net 2.0: Learning pdes from data with a numeric-symbolic hybrid deep network. *arXiv preprint arXiv:1812.04426*, 2018.

[88] Steven Brunton, Joshua Proctor, and Nathan Kutz. Sparse identification of nonlinear dynamics (sindy). In *APS Division of Fluid Dynamics Meeting Abstracts*, 2016.

[89] C. Soize. *Uncertainty Quantification: An Accelerated Course with Advanced Applications in Computational Engineering*. Interdisciplinary Applied Mathematics. Springer International Publishing, 2017.

[90] Sebastian Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, 2002.

[91] Michael S Hamada, Harry F Martz, C Shane Reese, and Alyson G Wilson. *Bayesian reliability*, volume 15. Springer, 2008.

[92] Phaedon-Stelios Koutsourelakis. Accurate uncertainty quantification using inaccurate computational models. *SIAM Journal on Scientific Computing*, 31(5):3274–3300, 2009.

[93] J. Kaipio and E. Somersalo. *Statistical and Computational Inverse Problems*. Applied Mathematical Sciences. Springer New York, 2006.

[94] Isabell Franck. *Sparse Variational Bayesian algorithms for large-scale inverse problems with applications in biomechanics*. PhD thesis, Technical University of Munich, 2017.

[95] A. Gelman, J.B. Carlin, H.S. Stern, D.B. Dunson, A. Vehtari, and D.B. Rubin. *Bayesian Data Analysis*. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis, 2013.

[96] M.J. Kochenderfer. *Decision Making Under Uncertainty: Theory and Application*. MIT Lincoln Laboratory Series. MIT Press, 2015.

[97] Phaedon-Stelios Koutsourelakis. Variational bayesian strategies for high-dimensional, stochastic design problems. *Journal of Computational Physics*, 308:124–152, 2016.

[98] Karl J Åström. *Introduction to stochastic control theory*. Courier Corporation, 2012.

[99] Christopher M Bishop. Model-based machine learning. *Phil. Trans. R. Soc. A*, 371(1984):20120222, 2013.

[100] Zoubin Ghahramani. Bayesian non-parametrics and the probabilistic approach to modelling. *Phil. Trans. R. Soc. A*, 371(1984):20110553, 2013.

[101] Samuel J Gershman and David M Blei. A tutorial on bayesian nonparametric models. *Journal of Mathematical Psychology*, 56(1):1–12, 2012.

[102] Subhashis Ghosal and Aad Van der Vaart. *Fundamentals of nonparametric Bayesian inference*, volume 44. Cambridge University Press, 2017.

[103] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[104] Michael Irwin Jordan. *Learning in graphical models*, volume 89. Springer Science & Business Media, 1998.

[105] Mathias Drton and Marloes H Maathuis. Structure learning in graphical modeling. *arXiv preprint arXiv:1606.02359*, 2016.

[106] Zhe Chen et al. Bayesian filtering: From kalman filters to particle filters, and beyond. *Statistics*, 182(1):1–69, 2003.

[107] M Yu Byron, Krishna V Shenoy, and Maneesh Sahani. Derivation of kalman filtering and smoothing equations. In *Technical report*. Stanford University, 2004.

[108] James Carpenter, Peter Clifford, and Paul Fearnhead. Improved particle filter for nonlinear problems. *IEE Proceedings-Radar, Sonar and Navigation*, 146(1):2–7, 1999.

[109] Nick Foti, Jason Xu, Dillon Laird, and Emily Fox. Stochastic variational inference for hidden markov models. *Advances in neural information processing systems*, 27, 2014.

[110] Zoubin Ghahramani and Geoffrey E. Hinton. Parameter estimation for linear dynamical systems. Technical report, 1996.

[111] Alfréd Rényi. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, volume 4, pages 547–562. University of California Press, 1961.

[112] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[113] Khadiga A.. Arwini and Christopher Terence John Dodson. *Information Geometry: Near Randomness and Near Independence*. Springer, 2008.

[114] N. Ay, J. Jost, H.V. Lê, and L. Schwachhöfer. *Information Geometry*. Ergebnisse der Mathematik und ihrer Grenzgebiete. 3. Folge / A Series of Modern Surveys in Mathematics. Springer International Publishing, 2017.

[115] Ariel Caticha. The basics of information geometry. In *AIP Conference Proceedings*, volume 1641, pages 15–26. AIP, 2015.

[116] MTCAJ Thomas and A Thomas Joy. *Elements of information theory*. Wiley-Interscience, 2006.

[117] Alexander Ly, Maarten Marsman, Josine Verhagen, Raoul PPP Grasman, and Eric-Jan Wagenmakers. A tutorial on fisher information. *Journal of Mathematical Psychology*, 80:40–55, 2017.

[118] C Radhakrishna Rao. Information and the accuracy attainable in the estimation of statistical parameters. In *Breakthroughs in statistics*, pages 235–247. Springer, 1992.

[119] Steven A Frank. Natural selection maximizes fisher information. *Journal of Evolutionary Biology*, 22(2):231–244, 2009.

[120] L Lorne Campbell. An extended čencov characterization of the information metric. *Proceedings of the American Mathematical Society*, 98(1):135–141, 1986.

[121] Shun-ichi Amari. Information geometry in optimization, machine learning and statistical inference. *Frontiers of Electrical and Electronic Engineering in China*, 5(3):241–260, 2010.

[122] Shun-ichi Amari. Divergence function, information monotonicity and information geometry. In *Workshop on information theoretic methods in science and engineering (WITMSE)*. Citeseer, 2009.

[123] Wikipedia. Kullback–Leibler divergence — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=Kullback%E2%80%93Leibler%20divergence&oldid=1102725879`, 2022. [Online; accessed 09-August-2022].

[124] Wikipedia. Mutual information — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=Mutual%20information&oldid=1102789521`, 2022. [Online; accessed 09-August-2022].

[125] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[126] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

[127] Ali Lotfi Rezaabad and Sriram Vishwanath. Learning representations by maximizing mutual information in variational autoencoders. In *2020 IEEE International Symposium on Information Theory (ISIT)*, pages 2729–2734. IEEE, 2020.

[128] Andrew M Stuart. Inverse problems: a bayesian perspective. *Acta numerica*, 19:451–559, 2010.

[129] Tan Bui-Thanh. Bayes is optimal. 07 2015.

[130] Arnold Zellner. Optimal information processing and bayes's theorem. *The American Statistician*, 42(4):278–280, 1988.

[131] Tan Bui-Thanh. The optimality of bayes' theorem, 2021.

[132] David JC MacKay. Probable networks and plausible predictions-a review of practical bayesian methods for supervised neural networks. *Network: computation in neural systems*, 6(3):469, 1995.

[133] David P. Wipf and Srikantan S. Nagarajan. A new view of automatic relevance determination. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1625–1632. Curran Associates, Inc., 2008.

[134] Anqi Wu, Mijung Park, Oluwasanmi O Koyejo, and Jonathan W Pillow. Sparse bayesian structure learning with "dependent relevance determination" priors. *Advances in Neural Information Processing Systems*, 27, 2014.

[135] Mike West and Michael D Escobar. *Hierarchical priors and mixture models, with application in regression and density estimation.* Institute of Statistics and Decision Sciences, Duke University, 1993.

[136] Rajesh Ranganath, Dustin Tran, and David Blei. Hierarchical variational models. In *International Conference on Machine Learning*, pages 324–333, 2016.

[137] Peter J Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

[138] Richard J Boys and Daniel A Henderson. A bayesian approach to dna sequence segmentation. *Biometrics*, 60(3):573–581, 2004.

[139] Simo Särkkä. *Bayesian filtering and smoothing*. Number 3. Cambridge university press, 2013.

[140] Jan-Willem van de Meent, Brooks Paige, Hongseok Yang, and Frank Wood. An introduction to probabilistic programming. *arXiv preprint arXiv:1809.10756*, 2018.

[141] Eli Bingham, Jonathan P. Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D. Goodman. Pyro: Deep Universal Probabilistic Programming. *Journal of Machine Learning Research*, 2018.

[142] Dustin Tran, Matthew D Hoffman, Rif A Saurous, Eugene Brevdo, Kevin Murphy, and David M Blei. Deep probabilistic programming. *arXiv preprint arXiv:1701.03757*, 2017.

[143] John Salvatier, Thomas V Wiecki, and Christopher Fonnesbeck. Probabilistic programming in python using pymc3. *PeerJ Computer Science*, 2:e55, 2016.

[144] Bob Carpenter, Andrew Gelman, Matthew D Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. *Journal of statistical software*, 76(1), 2017.

[145] Robert E Kass and Adrian E Raftery. Bayes factors. *Journal of the american statistical association*, 90(430):773–795, 1995.

[146] John K Kruschke. Bayesian estimation supersedes the t test. *Journal of Experimental Psychology: General*, 142(2):573, 2013.

[147] Jeffrey R. Stevens. Introduction to bayes factors, May 2019.

[148] Homer H Dubs. The principle of insufficient reason. *Philosophy of Science*, 9(2):123–131, 1942.

[149] Lewis Smith. A gentle introduction to information geometry, Sep 2019.

[150] José M Bernardo and Adrian FM Smith. *Bayesian theory*, volume 405. John Wiley & Sons, 2009.

[151] Jose M Bernardo. Reference posterior distributions for bayesian inference. *Journal of the Royal Statistical Society: Series B (Methodological)*, 41(2):113–128, 1979.

[152] James O Berger, José M Bernardo, and Dongchu Sun. The formal definition of reference priors. *The Annals of Statistics*, 37(2):905–938, 2009.

[153] Yansong Gao, Rahul Ramesh, and Pratik Chaudhari. Deep reference priors: What is the best way to pretrain a model? *arXiv preprint arXiv:2202.00187*, 2022.

[154] David John Cameron Mackay. *Bayesian methods for adaptive models*. PhD thesis, California Institute of Technology, 1992.

[155] Vijay Balasubramanian. Mdl, bayesian inference, and the geometry of the space of probability distributions. *Advances in minimum description length: Theory and applications*, pages 81–98, 2005.

[156] In Jae Myung, Vijay Balasubramanian, and Mark A Pitt. Counting probability distributions: Differential geometry and model selection. *Proceedings of the National Academy of Sciences*, 97(21):11170–11175, 2000.

[157] Vijay Balasubramanian. A geometric formulation of occam's razor for inference of parametric distributions. *arXiv preprint adap-org/9601001*, 1996.

[158] Jorma J Rissanen. Fisher information and stochastic complexity. *IEEE transactions on information theory*, 42(1):40–47, 1996.

[159] Hirotugu Akaike. Information theory and an extension of the maximum likelihood principle,[w:] proceedings of the 2nd international symposium on information, bn petrow, f. *Czaki, Akademiai Kiado, Budapest*, 1973.

[160] Gideon Schwarz. Estimating the dimension of a model. *The annals of statistics*, pages 461–464, 1978.

[161] Joshua Brett Tenenbaum. *A Bayesian framework for concept learning*. PhD thesis, Massachusetts Institute of Technology, 1999.

[162] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

[163] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic Variational Inference. *J. Mach. Learn. Res.*, 14(1):1303–1347, May 2013.

[164] Cheng Zhang, Judith Bütepage, Hedvig Kjellström, and Stephan Mandt. Advances in variational inference. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):2008–2026, 2018.

[165] C. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer Texts in Statistics. Springer New York, 2013.

[166] F. Liang, C. Liu, and R. Carroll. *Advanced Markov Chain Monte Carlo Methods: Learning from Past Samples*. Wiley Series in Computational Statistics. Wiley, 2011.

[167] Tim Salimans, Diederik Kingma, and Max Welling. Markov chain monte carlo and variational inference: Bridging the gap. In *International Conference on Machine Learning*, pages 1218–1226. PMLR, 2015.

[168] Achille Thin, Nikita Kotelevskii, Jean-Stanislas Denain, Leo Grinsztajn, Alain Durmus, Maxim Panov, and Eric Moulines. Metflow: A new efficient method for bridging the gap between markov chain monte carlo and variational inference. *arXiv preprint arXiv:2002.12253*, 2020.

[169] Antonios Alexos, Alex James Boyd, and Stephan Mandt. Structured stochastic gradient mcmc: a hybrid vi and mcmc approach. In *Fourth Symposium on Advances in Approximate Bayesian Inference*, 2021.

[170] Shun-ichi Amari, Shiro Ikeda, and Hidetoshi Shimokawa. Information geometry of α-projection in mean field approximation. *Advanced Mean Field Methods*, pages 241–257, 2001.

[171] Jose Hernandez-Lobato, Yingzhen Li, Mark Rowland, Thang Bui, Daniel Hernández-Lobato, and Richard Turner. Black-box alpha divergence minimization. In *International Conference on Machine Learning*, pages 1511–1520. PMLR, 2016.

[172] Yingzhen Li and Richard E Turner. Rényi divergence variational inference. *Advances in neural information processing systems*, 29, 2016.

[173] Adji Bousso Dieng, Dustin Tran, Rajesh Ranganath, John Paisley, and David Blei. Variational inference via χ upper bound minimization. *Advances in Neural Information Processing Systems*, 30, 2017.

[174] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan, 2017.

[175] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. *arXiv preprint arXiv:1711.01558*, 2017.

[176] Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Advances In Neural Information Processing Systems*, pages 2378–2386, 2016.

[177] Yingzhen Li and Yarin Gal. Dropout inference in bayesian neural networks with alpha-divergences. *arXiv preprint arXiv:1703.02914*, 2017.

[178] Peng Chen, Keyi Wu, Joshua Chen, Thomas O'Leary-Roseberry, and Omar Ghattas. Projected stein variational newton: A fast and scalable bayesian inference method in high dimensions. *arXiv preprint arXiv:1901.08659*, 2019.

[179] Rajesh Ranganath, Dustin Tran, Jaan Altosaar, and David Blei. Operator variational inference. *Advances in Neural Information Processing Systems*, 29, 2016.

[180] Rajesh Ranganath, Sean Gerrish, and David Blei. Black Box Variational Inference. In Samuel Kaski and Jukka Corander, editors, *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, volume 33 of *Proceedings of Machine Learning Research*, pages 814–822, Reykjavik, Iceland, 22–25 Apr 2014. PMLR.

[181] Jeffrey Regier, Michael I Jordan, and Jon McAuliffe. Fast black-box variational inference through stochastic trust-region optimization. In *Advances in Neural Information Processing Systems*, pages 2399–2408, 2017.

[182] Antti Honkela, Tapani Raiko, Mikael Kuusela, Matti Tornio, and Juha Karhunen. Approximate riemannian conjugate gradient learning for fixed-form variational bayes. *Journal of Machine Learning Research*, 11(Nov):3235–3268, 2010.

[183] Marcin Tomczak, Siddharth Swaroop, and Richard Turner. Efficient low rank gaussian variational inference for neural networks. *Advances in Neural Information Processing Systems*, 33:4610–4622, 2020.

[184] Ferenc Huszár. Variational inference using implicit distributions. *arXiv preprint arXiv:1702.08235*, 2017.

[185] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

[186] Christos Louizos and Max Welling. Multiplicative normalizing flows for variational bayesian neural networks. *arXiv preprint arXiv:1703.01961*, 2017.

[187] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

[188] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[189] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[190] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018.

[191] Xi-Lin Li. Preconditioned Stochastic Gradient Descent. *arXiv*, pages 1–12, 2015.

[192] Shakir Mohamed, Mihaela Rosca, Michael Figurnov, and Andriy Mnih. Monte carlo gradient estimation in machine learning. *J. Mach. Learn. Res.*, 21(132):1–62, 2020.

[193] Geoffrey Roeder, Yuhuai Wu, and David K Duvenaud. Sticking the landing: Simple, lower-variance gradient estimators for variational inference. In *Advances in Neural Information Processing Systems*, pages 6928–6937, 2017.

[194] Francisco R Ruiz, Michalis Titsias RC AUEB, and David Blei. The generalized reparameterization gradient. In *Advances in Neural Information Processing Systems*, pages 460–468, 2016.

[195] Yarin Gal and Zoubin Ghahramani. On modern deep learning and variational inference. In *Advances in Approximate Bayesian Inference workshop, NIPS*, volume 2, 2015.

[196] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[197] Edwin Thompson Jaynes. How does the brain do plausible reasoning? In *Maximum-entropy and Bayesian methods in science and engineering*, pages 1–24. Springer, 1988.

[198] Marc Toussaint. Probabilistic inference as a model of planned behavior. *Künstliche Intell.*, 23:23–29, 2009.

[199] Johan Ludwig William Valdemar Jensen. Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta mathematica*, 30(1):175–193, 1906.

[200] Matthew D Hoffman and Matthew J Johnson. Elbo surgery: yet another way to carve up the variational evidence lower bound. In *Workshop in Advances in Approximate Bayesian Inference, NIPS*, volume 1, 2016.

[201] Andreas C. Damianou, Michalis K. Titsias, and Neil D. Lawrence. Variational inference for latent variables and uncertain inputs in gaussian processes. *Journal of Machine Learning Research*, 17(42):1–62, 2016.

[202] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[203] Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

[204] Miin-Shen Yang, Chien-Yo Lai, and Chih-Ying Lin. A robust em clustering algorithm for gaussian mixture models. *Pattern Recognition*, 45(11):3950–3961, 2012.

[205] Radford M Neal and Geoffrey E Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pages 355–368. Springer, 1998.

[206] Julian Besag. Markov chain monte carlo for statistical inference. *Center for Statistics and the Social Sciences*, 9:24–25, 2001.

[207] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[208] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[209] Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential monte carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006.

[210] Mary Kathryn Cowles and Bradley P Carlin. Markov chain monte carlo convergence diagnostics: a comparative review. *Journal of the American Statistical Association*, 91(434):883–904, 1996.

[211] Vivekananda Roy. Convergence diagnostics for markov chain monte carlo. *Annual Review of Statistics and Its Application*, 7:387–412, 2020.

[212] Andrew Gelman, Walter R Gilks, and Gareth O Roberts. Weak convergence and optimal scaling of random walk metropolis algorithms. *The annals of applied probability*, 7(1):110–120, 1997.

[213] Andrew Gelman and Donald B Rubin. Inference from iterative simulation using multiple sequences. *Statistical science*, 7(4):457–472, 1992.

[214] Stephen P Brooks and Andrew Gelman. General methods for monitoring convergence of iterative simulations. *Journal of computational and graphical statistics*, 7(4):434–455, 1998.

[215] John F Geweke et al. Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments. Technical report, Federal Reserve Bank of Minneapolis, 1991.

[216] William A Link and Mitchell J Eaton. On thinning of chains in mcmc. *Methods in ecology and evolution*, 3(1):112–115, 2012.

[217] Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid monte carlo. *Physics letters B*, 195(2):216–222, 1987.

[218] Tatiana Xifara, Chris Sherlock, Samuel Livingstone, Simon Byrne, and Mark Girolami. Langevin diffusions and the metropolis-adjusted langevin algorithm. *Statistics & Probability Letters*, 91:14–19, 2014.

[219] Ben Calderhead. *Differential geometric MCMC methods and applications*. PhD thesis, University of Glasgow, 2011.

[220] Matthew D Hoffman, Andrew Gelman, et al. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623, 2014.

[221] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6):721–741, 1984.

[222] Radford M Neal. Markov chain monte carlo methods based onslicing'the density function. *Preprint*, 1997.

[223] Radford M Neal. Slice sampling. *Annals of statistics*, pages 705–741, 2003.

[224] Iain Murray, Ryan Adams, and David MacKay. Elliptical slice sampling. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 541–548. JMLR Workshop and Conference Proceedings, 2010.

[225] Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential monte carlo for bayesian computation. *Bayesian statistics*, 8(1):34, 2007.

[226] Nicholas G Polson and Vadim Sokolov. Deep learning: A bayesian perspective. *Bayesian Analysis*, 12(4):1275–1304, 2017.

[227] Andreas Damianou and Neil Lawrence. Deep gaussian processes. In *Artificial Intelligence and Statistics*, pages 207–215, 2013.

[228] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. *Advances in neural information processing systems*, 29, 2016.

[229] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[230] Thao Nguyen, Maithra Raghu, and Simon Kornblith. Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth. *arXiv preprint arXiv:2010.15327*, 2020.

[231] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[232] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[233] Gabriel Goh, Nick Cammarata †, Chelsea Voss †, Shan Carter, Michael Petrov, Ludwig Schubert, Alec Radford, and Chris Olah. Multimodal neurons in artificial neural networks. *Distill*, 2021. https://distill.pub/2021/multimodal-neurons.

[234] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[235] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 ieee information theory workshop (itw)*, pages 1–5. IEEE, 2015.

[236] Andrew G Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. *Advances in neural information processing systems*, 33:4697–4708, 2020.

[237] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[238] Eric Kauderer-Abrams. Quantifying translation-invariance in convolutional neural networks. *arXiv preprint arXiv:1801.01450*, 2017.

[239] Guillermo Valle-Perez, Chico Q Camargo, and Ard A Louis. Deep learning generalizes because the parameter-function map is biased towards simple functions. *arXiv preprint arXiv:1805.08522*, 2018.

[240] Chris Mingard, Joar Skalse, Guillermo Valle-Pérez, David Martínez-Rubio, Vladimir Mikulik, and Ard A Louis. Neural networks are a priori biased towards boolean functions with low entropy. *arXiv preprint arXiv:1909.11522*, 2019.

[241] Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

[242] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. *arXiv preprint arXiv:1711.00165*, 2017.

[243] Roman Novak, Lechao Xiao, Jaehoon Lee, Yasaman Bahri, Greg Yang, Jiri Hron, Daniel A Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Bayesian deep convolutional networks with many channels are gaussian processes. *arXiv preprint arXiv:1810.05148*, 2018.

[244] Greg Yang. Wide feedforward or recurrent neural networks of any architecture are gaussian processes. *Advances in Neural Information Processing Systems*, 32, 2019.

[245] Stephan Mandt, Matthew D Hoffman, and David M Blei. Stochastic gradient descent as approximate bayesian inference. *arXiv preprint arXiv:1704.04289*, 2017.

[246] Samuel L. Smith and Quoc V. Le. A Bayesian Perspective on Generalization and Stochastic Gradient Descent. *arXiv e-prints*, page arXiv:1710.06451, October 2017.

[247] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA, 20–22 Jun 2016. PMLR.

[248] Diederik P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems*, pages 2575–2583, 2015.

[249] Zhiting Hu, Zichao Yang, Ruslan Salakhutdinov, and Eric P Xing. On unifying deep generative models. *arXiv preprint arXiv:1706.00550*, 2017.

[250] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[251] Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.

[252] Markus Schöberl, Nicholas Zabaras, and Phaedon-Stelios Koutsourelakis. Predictive collective variable discovery with deep bayesian models. *The Journal of chemical physics*, 150(2):024109, 2019.

[253] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

[254] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[255] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

[256] Ivan Kobyzev, Simon JD Prince, and Marcus A Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):3964–3979, 2020.

[257] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

[258] Aaron Van Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International conference on machine learning*, pages 1747–1756. PMLR, 2016.

[259] Lucas Theis and Matthias Bethge. Ggenerative image modeling using spatial lstms. *Advances in Neural Information Processing Systems*, 28, 2015.

[260] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. *Advances in neural information processing systems*, 30, 2017.

[261] Apratim Bhattacharyya, Shweta Mahajan, Mario Fritz, Bernt Schiele, and Stefan Roth. Normalizing flows with multi-scale autoregressive priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8415–8424, 2020.

[262] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. In *International Conference on Machine Learning*, pages 2391–2400. PMLR, 2017.

[263] Shakir Mohamed and Balaji Lakshminarayanan. Learning in implicit generative models. *arXiv preprint arXiv:1610.03483*, 2016.

[264] O. Chapelle, B. Schölkopf, and A. Zien. Semi-supervised learning. *IEEE Transactions on Neural Networks*, 20(3), 2009.

[265] Andrei Atanov, Alexandra Volokhova, Arsenii Ashukha, Ivan Sosnovik, and Dmitry Vetrov. Semi-conditional normalizing flows for semi-supervised learning. *arXiv preprint arXiv:1905.00505*, 2019.

[266] Christopher M Bishop. Bayesian pca. *Advances in neural information processing systems*, pages 382–388, 1999.

[267] Shipeng Yu, Kai Yu, Volker Tresp, Hans-Peter Kriegel, and Mingrui Wu. Supervised probabilistic principal component analysis. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 464–473. ACM, 2006.

[268] Yoon Kim, Sam Wiseman, Andrew C Miller, David Sontag, and Alexander M Rush. Semi-amortized variational autoencoders. *arXiv preprint arXiv:1802.02550*, 2018.

[269] Chris Cremer, Xuechen Li, and David Duvenaud. Inference suboptimality in variational autoencoders. In *International Conference on Machine Learning*, pages 1078–1086. PMLR, 2018.

[270] Rogan Morrow and Wei-Chen Chiu. Variational autoencoders with normalizing flow decoders. *arXiv preprint arXiv:2004.05617*, 2020.

[271] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Balancing learning and inference in variational autoencoders. In *Proceedings of the aaai conference on artificial intelligence*, volume 33, pages 5885–5892, 2019.

[272] Masatoshi Uehara, Issei Sato, Masahiro Suzuki, Kotaro Nakayama, and Yutaka Matsuo. Generative adversarial nets from a density ratio estimation perspective. *arXiv preprint arXiv:1610.02920*, 2016.

[273] A. Papoulis and S.U. Pillai. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill series in electrical engineering: Communications and signal processing. McGraw-Hill, 2002.

[274] Salomon Bochner, Monotonic Functions, Stieltjes Integrals, Harmonic Analysis, Morris Tenenbaum, and Harry Pollard. *Lectures on Fourier Integrals. (AM-42)*. Princeton University Press, 1959.

[275] B. Hu and W. Schiehlen. On the simulation of stochastic processes by spectral representation. *Probabilistic Engineering Mechanics*, 12(2):105–113, 1997.

[276] Masanobu Shinozuka and George Deodatis. Simulation of Multi-Dimensional Gaussian Stochastic Fields by Spectral Representation. *Applied Mechanics Reviews*, 49(1):29–53, 01 1996.

[277] Raphael Sternfels and Phaedon-Stelios Koutsourelakis. Stochastic design and control in random heterogeneous materials. *International Journal for Multiscale Computational Engineering*, 9(4), 2011.

[278] Andrew Wilson and Ryan Adams. Gaussian process kernels for pattern discovery and extrapolation. In *International conference on machine learning*, pages 1067–1075. PMLR, 2013.

[279] Clive AJ Fletcher. Computational galerkin methods. In *Computational galerkin methods*, pages 72–85. Springer, 1984.

[280] Anders Logg, Kent-Andre Mardal, and Garth Wells. *Automated solution of differential equations by the finite element method: The FEniCS book*, volume 84. Springer Science & Business Media, 2012.

[281] Martin Alnæs, Jan Blechta, Johan Hake, August Johansson, Benjamin Kehlet, Anders Logg, Chris Richardson, Johannes Ring, Marie E Rognes, and Garth N Wells. The fenics project version 1.5. *Archive of Numerical Software*, 3(100), 2015.

[282] M. Raissi, P. Perdikaris, and G. Karniadakis. Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations. *arxiv e-print*, 2017.

[283] M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686 – 707, 2019.

[284] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations. *Journal of Computational Physics*, November 2018.

[285] Weinan E and Bing Yu. The Deep Ritz Method: A Deep Learning-Based Numerical Algorithm for Solving Variational Problems. *Communications in Mathematics and Statistics*, 6(1):1–12, March 2018.

[286] Jim Magiera, Deep Ray, Jan S Hesthaven, and Christian Rohde. Constraint-aware neural networks for riemann problems. *Journal of Computational Physics*, 409:109345, 2020.

[287] Jens Berg and Kaj Nyström. A unified deep artificial neural network approach to partial differential equations in complex geometries. *Neurocomputing*, 317:28–41, 2018.

[288] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Machine learning of linear differential equations using gaussian processes. *Journal of Computational Physics*, 348:683–693, 2017.

[289] Maximilian Rixner and Phaedon-Stelios Koutsourelakis. A probabilistic generative model for semi-supervised training of coarse-grained surrogates and enforcing physical constraints through virtual observables. *Journal of Computational Physics*, 434:110218, 2021.

[290] Julia Ling, Reese Jones, and Jeremy Templeton. Machine learning strategies for systems with invariance properties. *Journal of Computational Physics*, 318:22–35, 2016.

[291] Julia Ling, Andrew Kurzawski, and Jeremy Templeton. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics*, 807:155–166, 2016.

[292] Nils Thuerey, Philipp Holl, Maximilian Mueller, Patrick Schnell, Felix Trost, and Kiwon Um. *Physics-based Deep Learning*. WWW, 2021.

[293] Kiwon Um, Robert Brand, Yun Raymond Fei, Philipp Holl, and Nils Thuerey. Solver-in-the-loop: Learning from differentiable physics to interact with iterative pde-solvers. *Advances in Neural Information Processing Systems*, 33:6111–6122, 2020.

[294] Philipp Holl, Vladlen Koltun, and Nils Thuerey. Learning to control pdes with differentiable physics. *arXiv preprint arXiv:2001.07457*, 2020.

[295] Yuanming Hu, Luke Anderson, Tzu-Mao Li, Qi Sun, Nathan Carr, Jonathan Ragan-Kelley, and Frédo Durand. Difftaichi: Differentiable programming for physical simulation. *arXiv preprint arXiv:1910.00935*, 2019.

[296] Junbang Liang and Ming C Lin. Differentiable physics simulation. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*, 2020.

[297] Filipe de Avila Belbute-Peres, Kevin Smith, Kelsey Allen, Josh Tenenbaum, and J Zico Kolter. End-to-end differentiable physics for learning and control. *Advances in neural information processing systems*, 31:7178–7189, 2018.

[298] Yi-Ling Qiao, Junbang Liang, Vladlen Koltun, and Ming C Lin. Scalable differentiable physics for learning and control. *arXiv preprint arXiv:2007.02168*, 2020.

[299] Carlos A. Michelén Ströfer and Heng Xiao. End-to-end differentiable learning of turbulence models from indirect observations. *Theoretical and Applied Mechanics Letters*, 11(4):100280, 2021.

[300] Uwe Naumann. *The art of differentiating computer programs: an introduction to algorithmic differentiation*, volume 24. Siam, 2012.

[301] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[302] William Moses and Valentin Churavy. Instead of rewriting foreign code for machine learning, automatically synthesize fast gradients. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12472–12485. Curran Associates, Inc., 2020.

[303] William S Moses, Valentin Churavy, Ludger Paehler, Jan Hückelheim, Sri Hari Krishna Narayanan, Michel Schanen, and Johannes Doerfert. Reverse-mode automatic differentiation and optimization of gpu kernels via enzyme. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16, 2021.

[304] Andreas Look, Simona Doneva, Melih Kandemir, Rainer Gemulla, and Jan Peters. Differentiable implicit layers. *CoRR*, abs/2010.07078, 2020.

[305] Kenji Kawaguchi. On the theory of implicit deep learning: Global convergence with implicit layers. In *International Conference on Learning Representations*, 2021.

[306] Yin Cheng Ng, Nicolò Colombo, and Ricardo Silva. Bayesian semi-supervised learning with graph gaussian processes. *Advances in Neural Information Processing Systems*, 31, 2018.

[307] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[308] Xiaolin Li, Yichi Zhang, He Zhao, Craig Burkhart, L Catherine Brinson, and Wei Chen. A transfer learning approach for microstructure reconstruction and structure-property predictions. *Scientific reports*, 8(1):1–13, 2018.

[309] Nicholas Lubbers, Turab Lookman, and Kipton Barros. Inferring low-dimensional microstructure representations using convolutional neural networks. *Physical Review E*, 96(5):052111, 2017.

[310] A Kumar, L Nguyen, M DeGraef, and V Sundararaghavan. A markov random field approach for microstructure synthesis. *Modelling and Simulation in Materials Science and Engineering*, 24(3):035015, 2016.

[311] Ruijin Cang, Hechao Li, Hope Yao, Yang Jiao, and Yi Ren. Improving direct physical properties prediction of heterogeneous materials from imaging data via convolutional neural network and a morphology-aware generative model. *Computational Materials Science*, 150:212–221, 2018.

[312] Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, Yizhou Yu, and Mingli Song. Neural style transfer: A review. *IEEE transactions on visualization and computer graphics*, 26(11):3365–3385, 2019.

[313] Chuan Li and Michael Wand. Combining markov random fields and convolutional neural networks for image synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2479–2486, 2016.

[314] Leon Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. *Advances in neural information processing systems*, 28, 2015.

[315] Brian L DeCost, Toby Francis, and Elizabeth A Holm. Exploring the microstructure manifold: image texture representations applied to ultrahigh carbon steel microstructures. *Acta Materialia*, 133:30–40, 2017.

[316] Zijiang Yang, Yuksel C Yabansu, Reda Al-Bahrani, Wei-keng Liao, Alok N Choudhary, Surya R Kalidindi, and Ankit Agrawal. Deep learning approaches for mining structure-property linkages in high contrast composites from simulation datasets. *Computational Materials Science*, 151:278–287, 2018.

[317] Ahmet Cecen, Hanjun Dai, Yuksel C Yabansu, Surya R Kalidindi, and Le Song. Material structure-property linkages using three-dimensional convolutional neural networks. *Acta Materialia*, 146:76–84, 2018.

[318] Patxi Fernandez-Zelaia, Yuksel C Yabansu, and Surya R Kalidindi. A comparative study of the efficacy of local/global and parametric/nonparametric machine learning methods for establishing structure–property linkages in high-contrast 3d elastic composites. *Integrating Materials and Manufacturing Innovation*, 8(2):67–81, 2019.

[319] Haiyi Wu, Wen-Zhen Fang, Qinjun Kang, Wen-Quan Tao, and Rui Qiao. Predicting effective diffusivity of porous media from images by deep learning. *Scientific reports*, 9(1):1–12, 2019.

[320] J Zhou, G Cui, Z Zhang, C Yang, Z Liu, and M Sun. Graph neural networks: A review of methods andapplications. *arXiv preprint arXiv: 1812.08434*, 2018.

[321] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. *Advances in neural information processing systems*, 29, 2016.

[322] Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. Variational dropout sparsifies deep neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2498–2507. JMLR. org, 2017.

[323] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *International Conference on Machine Learning*, pages 1183–1192. PMLR, 2017.

[324] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

[325] Andreas Kirsch, Joost Van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. *Advances in neural information processing systems*, 32, 2019.

[326] Adam Foster. *Variational, Monte Carlo and Policy-Based Approaches to Bayesian Experimental Design*. PhD thesis, University of Oxford, 2022.

[327] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

[328] Saeed Masoudnia and Reza Ebrahimpour. Mixture of experts: a literature survey. *Artificial Intelligence Review*, 42(2):275–293, 2014.

[329] Xin Wang, Fisher Yu, Lisa Dunlap, Yi-An Ma, Ruth Wang, Azalia Mirhoseini, Trevor Darrell, and Joseph E Gonzalez. Deep mixture of experts via shallow embedding. In *Uncertainty in artificial intelligence*, pages 552–562. PMLR, 2020.

[330] Gang Bao, Xiaojing Ye, Yaohua Zang, and Haomin Zhou. Numerical solution of inverse problems by weak adversarial networks. *Inverse Problems*, 36(11):115003, 2020.

[331] Teresa Portone and Robert D Moser. Bayesian inference of an uncertain generalized diffusion operator. *SIAM/ASA Journal on Uncertainty Quantification*, 10(1):151–178, 2022.

[332] Steven Atkinson. Bayesian hidden physics models: Uncertainty quantification for discovery of nonlinear partial differential operators from data. *arXiv preprint arXiv:2006.04228*, 2020.

[333] Chris Lattner and Vikram Adve. LLVM: A compilation framework for lifelong program analysis and transformation. pages 75–88, San Jose, CA, USA, Mar 2004.

[334] PG Martinsson and JA Tropp. Randomized numerical linear algebra: foundations & algorithms (2020). *arXiv preprint arXiv:2002.01387*.

[335] Simon Bartels. *Probabilistic Linear Algebra*. PhD thesis, Eberhard Karls Universität Tübingen, 2021.

[336] Rodney Hill. On constitutive macro-variables for heterogeneous solids at finite strain. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 326(1565):131–147, 1972.

[337] Christian Miehe and Andreas Koch. Computational micro-to-macro transitions of discretized microstructures undergoing small strains. *Archive of Applied Mechanics*, 72(4):300–317, 2002.

[338] Jeremy Bleyer. *Numerical Tours of Computational Mechanics with FEniCS*, 2018.

A

Micro-to-Macro transition

In the following, we discuss the characterization of microstructures in terms of their effective macroscopic properties. The ensuant discussion is based on Hill's averaging theorem [336], which itself is predicated on an energy argument. In our context, we seek to determine effective macroscopic material properties for a representative volume element (RVE) of binary two-phase random media (generated by a stochastic process), such that the dissipated energy given its microscopic and effective macroscopic description becomes identical (see Figure A.11. As this necessitates a sufficiently fine spatial resolution to capture the variability of the microstructure, establishing the micro to macro transition (*coarse graining*) defines a numerically expensive operation. An abridged version of the exposition in this section can be found in the supplementary material of [31] (corresponding to section 3.2 in this thesis).

Structural Properties

In the following, we exemplarily consider the case where one is interested in the physical response of the microstructure under *structural* loading, assuming linear elastic material behavior on the microscopic scale. We will consider microstructures as realizations of random fields on our representative volume element $\mathcal{V} \subset \mathbb{R}^d$ (with its volume given by $|\mathcal{V}|$). On a microscopic scale, our assumptions then equate to the balance equation (BE) and constitutive equation (CE)

$$\text{(BE):} \qquad \operatorname{div}(\sigma) = 0 \qquad \forall\ s \in \mathcal{V} \tag{A.1}$$

$$\text{(CE):} \qquad \sigma = \mathbb{C}(s) : \epsilon \qquad \forall\ s \in \mathcal{V} \tag{A.2}$$

where σ and ϵ define the microscopic Cauchy stress and microscopic (infinitesimal) strain tensor, respectively. For binary two-phase random media the spatially variable fourth-order stiffness tensor $\mathbb{C}(s)$ takes only take two distinct values corresponding to the two binary phases $\mathcal{V}^{(0)} \subset \mathcal{V}, \mathcal{V}^{(1)} \subset \mathcal{V}$, with $\mathcal{V}^{(0)} \cap \mathcal{V}^{(1)} = \emptyset$ and $\mathcal{V}^0 \cup \mathcal{V}^{(1)} = \mathcal{V}$:

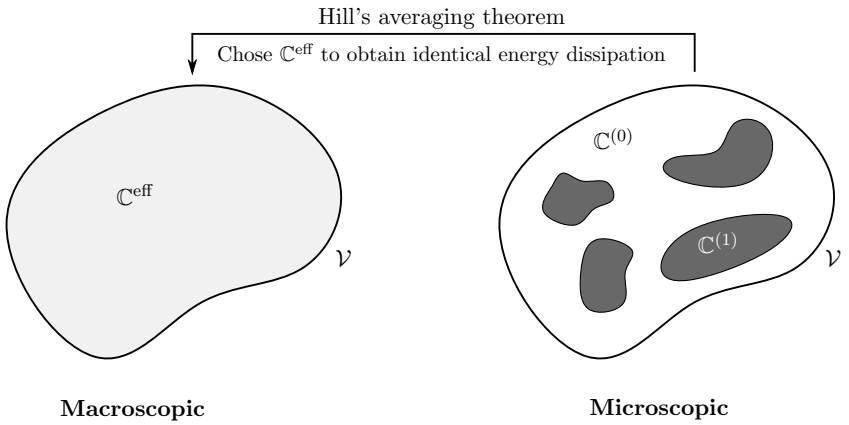


Figure A.1: The macroscopic stress $\Sigma = \langle\sigma\rangle$ and strain $E = \langle\epsilon\rangle$ are defined as spatial averages of microscopic stress σ and strain ϵ, related to each other by the effective, tangent elastic modulus $\mathbb{C}^{\text{eff}} = \partial_E \Sigma$. The effective material properties $\mathbb{C}^{\text{eff}}$ are determined by necessitating identical energy dissipation on both microscopic and macroscopic scales.

$$\mathbb{C}(s) = \begin{cases} \mathbb{C}^1 & \text{for } s \in \mathcal{V}^{(1)} \\ \mathbb{C}^0 & \text{for } s \in \mathcal{V}^{(0)} \end{cases} \tag{A.3}$$

Under the assumption of isotropic and homogenous media, both $\mathbb{C}^0$ and $\mathbb{C}^1$ are fully characterized by their respective Lamé parameters. We define the macroscopic stress $\Sigma = \langle\sigma\rangle$ as the *spatial* average $\Sigma = |\mathcal{V}|^{-1} \int \sigma \, \mathrm{d}V$. We can furthermore relate the macroscopic stress to the boundary tractions t by means of Stokes' theorem, i.e. for any tensor T_{ijk} it holds $\int_{\mathcal{V}} \partial_i T_{ijk} dV = \oint_{\partial\mathcal{V}} n_i T_{ijk} \, \mathrm{d}A$, and as such

$$\begin{aligned}
[\Sigma]_{ik} &= \left[\frac{1}{|\mathcal{V}|} \oint_{\partial\mathcal{V}} \operatorname{sym}(t \otimes s) \, \mathrm{d}A \right]_{ik} \\
&= \frac{1}{|\mathcal{V}|} \oint_{\partial\mathcal{V}} \frac{1}{2} \left(\sigma_{ij} n_j s_k + \sigma_{kj} n_k s_i \right) \mathrm{d}A \\
&= \frac{1}{|\mathcal{V}|} \frac{1}{2} \int \frac{1}{2} \left(\sigma_{ij} s_k \right)_{,j} + \left(\sigma_{kj} s_i \right)_{,j} \mathrm{d}V \\
&= \frac{1}{|\mathcal{V}|} \frac{1}{2} \int \sigma_{ij,j} s_k + \sigma_{ij} s_{k,j} + \sigma_{kj,j} s_i + \sigma_{kj} s_{i,j} \, \mathrm{d}V \\
&= \frac{1}{|\mathcal{V}|} \frac{1}{2} \int \sigma_{ij} \delta_{kj} + \sigma_{kj} \delta_{ij} \, \mathrm{d}V \\
&= \frac{1}{|\mathcal{V}|} \int \sigma_{ik} \, \mathrm{d}V = \left[\frac{1}{|\mathcal{V}|} \int \sigma \, \mathrm{d}V \right]_{ik}
\end{aligned} \tag{A.4}$$

with spatial dimension d, tractions $t \in \mathbb{R}^d$, normal vector $n \in \mathbb{R}^d$, spatial coordinates $s \in \mathcal{V}$ and the closed boundary of the domain $\partial\mathcal{V}$. Similarly, the macroscopic strain E corresponding to a second order tensor is given as the spatial average of the microscopic strain ϵ

$$\begin{aligned} E &= \frac{1}{|\mathcal{V}|} \int \operatorname{sym}(u \otimes n) \, \mathrm{d}A \\ &= \frac{1}{|\mathcal{V}|} \int \frac{1}{2} (u_{i,j} + u_{j,i}) \, \mathrm{d}V = \frac{1}{|\mathcal{V}|} \int \epsilon \, \mathrm{d}V \end{aligned} \tag{A.5}$$

The effective physical properties of the material is characterized by an effective tangent elastic modulus $\mathbb{C}^{\text{eff}} := \partial_E \Sigma$, i.e., a fourth-order tensor linearly relating macroscopic stress Σ and strain E for the RVE

$$\Sigma = \mathbb{C}^{\text{eff}} : E \tag{A.6}$$

The macroscopic, effective behavior characterized by $\mathbb{C}^{\text{eff}}$ is implicitly defined by the constraint that Eq. (A.6) satisfies Hill's averaging theorem, i.e., the energy dissipated according to the macroscopic description is equal to the energy dissipated according to the microscopic displacements u and the boundary tractions t:

$$\Sigma : E = \frac{1}{|\mathcal{V}|} \int_{\partial\mathcal{V}} t \cdot u \, \mathrm{d}A \tag{A.7}$$

If we can identify (for $d = 2$) the macroscopic stress Σ satisfying this requirement for the following elementary macroscopic strain modes corresponding to pure tension or shear states

$$\hat{E}^{(1)} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad \hat{E}^{(2)} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \quad \hat{E}^{(3)} = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \tag{A.8}$$

then obviously from Eq. (A.6), this enables identification of the effective tangent elastic modulus $\mathbb{C}^{\text{eff}}$ defining the effective macroscopic behavior of the microstructure under structural loading. As detailed in [337], one of several approaches to numerically identify $\mathbb{C}^{\text{eff}}$ in compliance with Hill's averaging theorem is to introduce periodic boundary conditions and augment the constitutive equation (A.2) and balance equation (A.1) with a set of additional equations

$$\epsilon = \hat{E}^{(c)} + \nabla_s v \qquad \text{in } \mathcal{V} \tag{A.9}$$

$$v \qquad \text{is } \mathcal{V}\text{-periodic} \tag{A.10}$$

$$t = \sigma \cdot n \qquad \text{is } \mathcal{V}\text{-antiperiodic} \tag{A.11}$$

for $c \in \{1, 2, 3\}$, where v defines a periodic (microscopic) fluctuation across the domain $\mathcal{V}$, and t being the anti-periodic (microscopic) boundary tractions. The solution of this set of partial differential equations under the given boundary conditions can, e.g., be rephrased in weak form using Galerkin's method. This suggests to introduce a discrete function space $\mathbb{V}_h$ for v and to simultaneously solve for the periodic fluctuation $v \in \mathbb{V}_h$ and Lagrange multiplier $\lambda \in \mathbb{R}^d$ such that [338]

$$\int_{\mathcal{V}} \left(\hat{E}^{(c)} + \nabla_s v\right) : \mathbb{C}\,(s) : \nabla_s \hat{v}\, \mathrm{d}\mathcal{V} + \int_{\mathcal{V}} \lambda \cdot \hat{v}\, \mathrm{d}\mathcal{V} + \int_{\mathcal{V}} \hat{\lambda} \cdot v\, \mathrm{d}\mathcal{V} = 0 \tag{A.12}$$

for all $(\hat{v}, \hat{\lambda}) \in \mathbb{V}_h \times \mathbb{R}^2$ acting as test functions. Here the Lagrange multiplier λ has been introduced to disambiguate with respect to rigid body modes. In summary this suggests the following approach: the set of coupled differential equations under periodic boundary conditions are solved for the three elementary load cases $\hat{E}^{(c)}, c \in \{1, 2, 3\}$, and the associated macroscopic stress Σ is inferred (see (A.4)). Knowledge of the macroscopic stress for the elementary load cases straightforwardly enables to determine $\mathbb{C}^{\text{eff}}$ given Eq. (A.6). In closing, we note that the discussion can analogously be repeated for effective thermal properties of the microstructures (corresponding to the diffusion process defined by Eqs. (2.51) and (2.52)). In this setting a temperature field u and flux q have as counterpart the macroscopic temperature field U and flux Q, which are connected by Fourier's law, i.e., $Q = -a^{\text{eff}}\,(\nabla U)$, with a^{eff} a symmetric positive definite tensor relating temperature gradient and flux. The effective conductivity tensor a^{eff} is then similarly determined utilizing Hill's averaging theorem, i.e., postulating identical energy dissipation on the microscopic and macroscopic scale, and solving a coupled set of partial differential equations.

B
Paper A

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

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

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

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

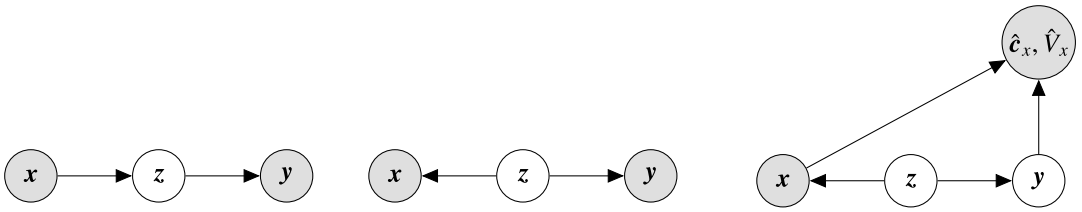
with $\boldsymbol{\theta}$ being the trainable parameters of the model. The variables $\boldsymbol{z}\in\mathbb{R}^Q$ represent the lower-dimensional (i.e. $Q << d_x, d_y$) information bottleneck between inputs and outputs. In the aforementioned works, these have been associated with a lower-fidelity 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.

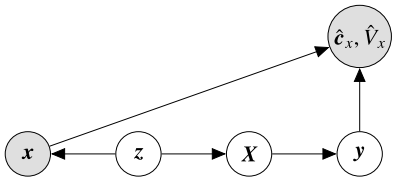


Fig. 2. Node $\boldsymbol{X}$ corresponds to the inputs of a deterministic coarse-grained model (CGM), implying that $\boldsymbol{z}$ is encouraged not only to learn a representation of the inputs $\boldsymbol{x}$, but also features that through the CGM can be predictive of the FGM output $\boldsymbol{y}$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

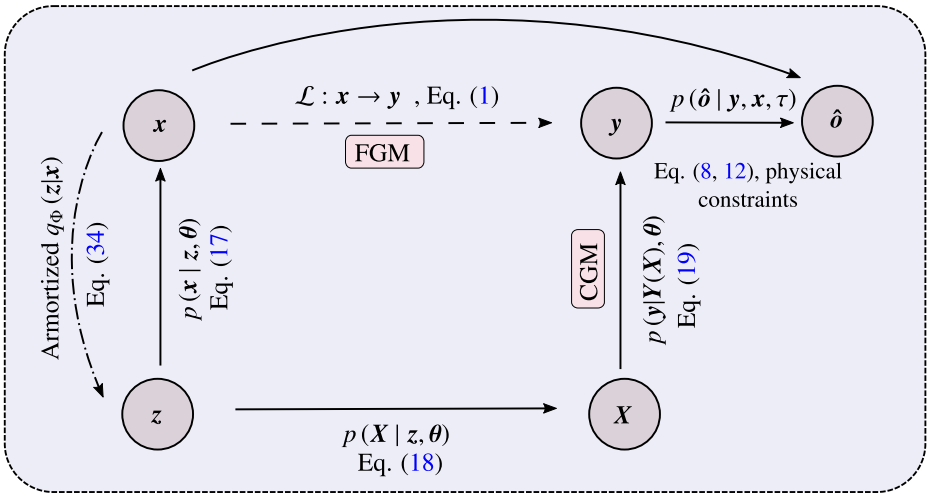


Fig. 3. A schematic overview of the building blocks of the generative model. All solid black arrows correspond to the conditional densities Eq. (17)–(19), i.e. encode conditional dependence assumptions, and therefore define the joint distribution $p_{\boldsymbol{\theta}}\left(\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{X}, \boldsymbol{y}, \hat{\boldsymbol{o}}\right) = p(\boldsymbol{z})\, p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\theta})\, p(\boldsymbol{X}|\boldsymbol{z}, \boldsymbol{\theta})\, p(\boldsymbol{y}|\boldsymbol{Y}(\boldsymbol{X}), \boldsymbol{\theta})\, p\left(\hat{\boldsymbol{o}}|\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\tau}\right)$. 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 $\boldsymbol{\theta} = \left\{\boldsymbol{\theta}_x, \boldsymbol{\theta}_g, \boldsymbol{\theta}_y, \boldsymbol{S}_X, \boldsymbol{S}_y\right\}$ 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 $q_{\boldsymbol{\xi}}$ that factorizes as follows

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

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 $\{\mathcal{Z}_l,\mathcal{X}_l\}$ and $\{\mathcal{Z}_{\mathcal{O}},\mathcal{X}_{\mathcal{O}},\mathcal{Y}_{\mathcal{O}}\}$. Furthermore, $q\left(\mathcal{R}\right)$ 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 $q_{\boldsymbol{\xi}}\left(\boldsymbol{\theta}\right)$). Given this, the ELBO becomes:

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

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

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

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

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

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

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

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

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 $\boldsymbol{\epsilon}_{(k)} \sim p(\boldsymbol{\epsilon})$, $k = 1, ..., K$;
- $\mathcal{R}_{(k)} \leftarrow \varrho_{\xi}^{\mathcal{R}}\left(\boldsymbol{\epsilon}_{(k)}\right)$ $\quad \boldsymbol{\theta}_{(k)} \leftarrow \varrho_{\xi}^{\boldsymbol{\theta}}\left(\boldsymbol{\epsilon}_{(k)}\right)$ $\quad k = 1, ..., K$;
- `// Monte Carlo estimate of ELBO`
- Estimate $\hat{\mathcal{F}} \leftarrow \sum_{k=1}^{K} \mathcal{F}\left(\boldsymbol{\theta}_{(k)}, \mathcal{R}_{(k)}\right)$; `// Equation (28)`
- `// Backpropagate`
- $\boldsymbol{g}_{\xi} \leftarrow \nabla_{\xi} \sum_{k=1}^{K} \mathcal{F}\left(\boldsymbol{\theta}_{(k)}, \mathcal{R}_{(k)}\right)$;
- `// 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:

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

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 $q_{\Phi}\left(\boldsymbol{z}^{(i_u)}|\boldsymbol{x}^{(i_u)}\right)$ [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

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

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 $p_{\theta}(\boldsymbol{z}|\boldsymbol{x})$ 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
  $q^*(\boldsymbol{z}) \leftarrow \arg\max_\zeta \hat{\mathcal{F}}_u\left(q_\zeta(\boldsymbol{z})\right)$; // Equation (37)
end
for $k \leftarrow 1$ **to** K **do**
  Sample $\boldsymbol{z}^{(k)} \sim q^*(\boldsymbol{z})$;
  Sample $\boldsymbol{X}^{(k)} \sim p\left(\boldsymbol{X}|\boldsymbol{z}^{(k)}, \boldsymbol{\theta}_{MAP}\right)$; // Equation (18)
  Sample $\boldsymbol{y}^{(k)} \sim p\left(\boldsymbol{y}|\boldsymbol{X}^{(k)}, \boldsymbol{\theta}_{MAP}\right)$; // Equation (19)
end
Construct sample-based approximation $\tilde{p}(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D})$ using samples $\boldsymbol{y}^{(k)}, k = 1, ..., K$

We combine the noisy estimates of the gradient $\nabla_\xi \mathcal{F}$ 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 $q_\xi(\boldsymbol{\theta})$ 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 $p_\theta(\boldsymbol{x}, \boldsymbol{y})$, 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 $q_\xi(\boldsymbol{\theta}) = \delta(\boldsymbol{\theta} - \theta_{MAP})$ 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_ξ 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* $p\left(\boldsymbol{y}|\boldsymbol{x}, \hat{\boldsymbol{o}}, \mathcal{D}\right) \propto p\left(\hat{\boldsymbol{o}}|\boldsymbol{y}, \boldsymbol{x}\right) p(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D})$ 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.

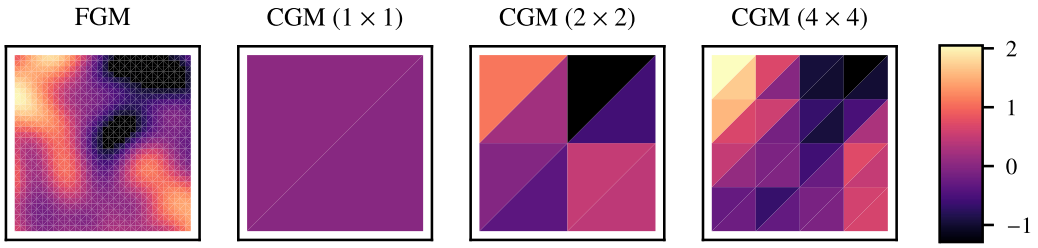


Fig. 4. Comparison of a sample $\boldsymbol{x}^{(i)}$ of the discretized of the Gaussian random field $\lambda(\boldsymbol{s})$ of the FGM (left - Equation (48) with $l_\lambda = 0.15$) with the (log of the posterior mean of the) corresponding $\boldsymbol{X}^{(i)}$ for three different CGM discretizations, 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.

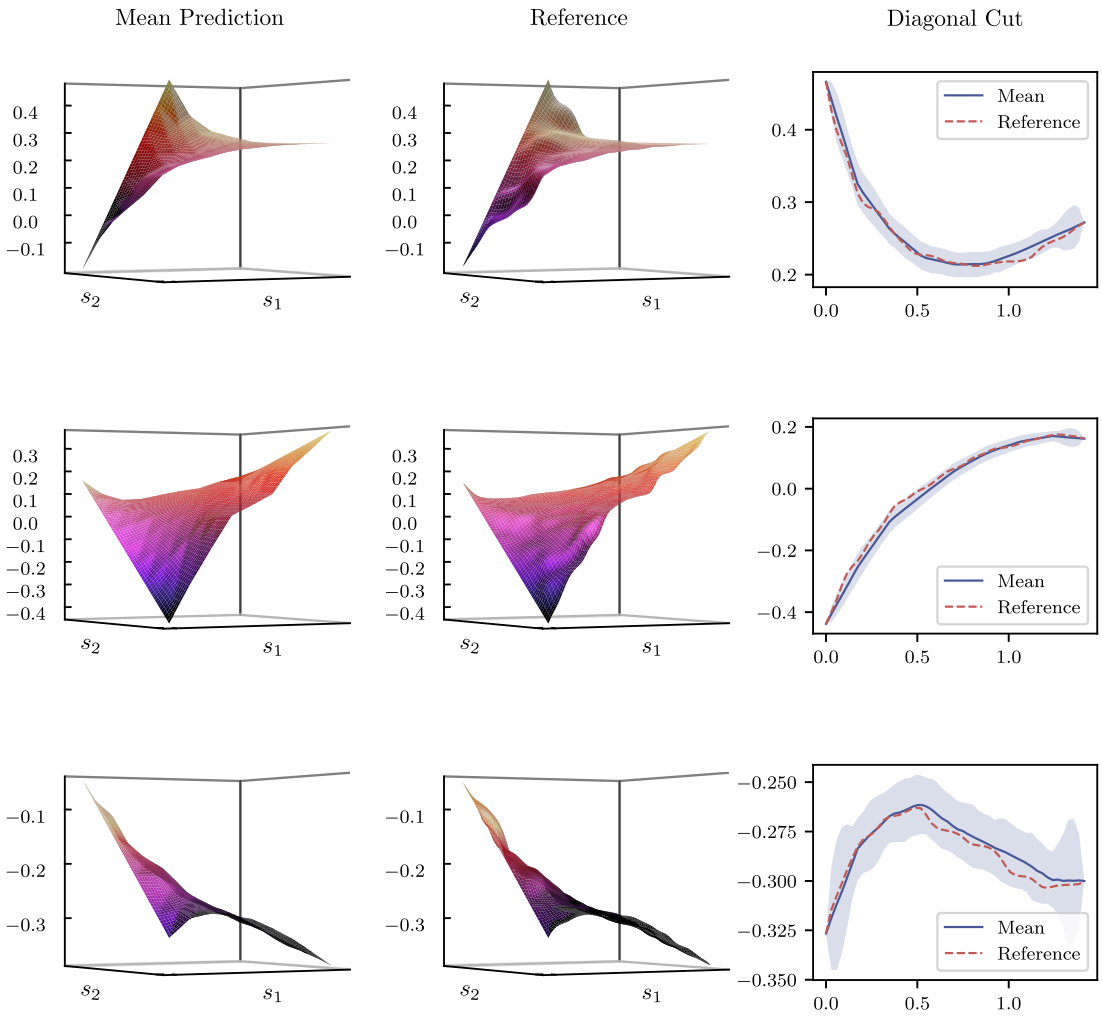


Fig. 5. The *left* column provides examples of the mean of the predictive posterior $p(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D})$ for various $\boldsymbol{x}$ not seen during training. The *middle* column contains the actual output $\boldsymbol{y}$ obtained by solving the FGM (ground truth / reference). 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.

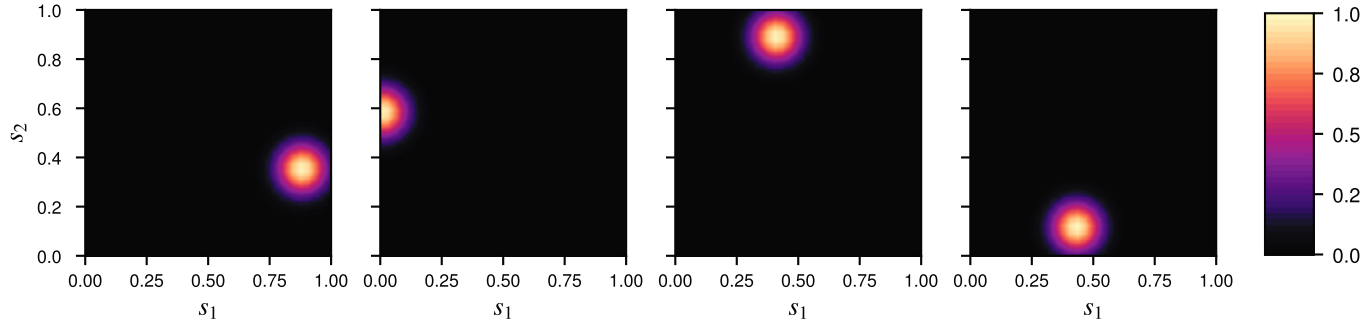


Fig. 6. Illustration of 4 randomly sampled radial basis-type weight functions (Eq. (55)) corresponding to the Randomized Residuals. Instead of using collocation points at which the PDE is enforced, we randomly sample Galerkin weight functions that enforce governing equations in a spatially-averaged sense.

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*.

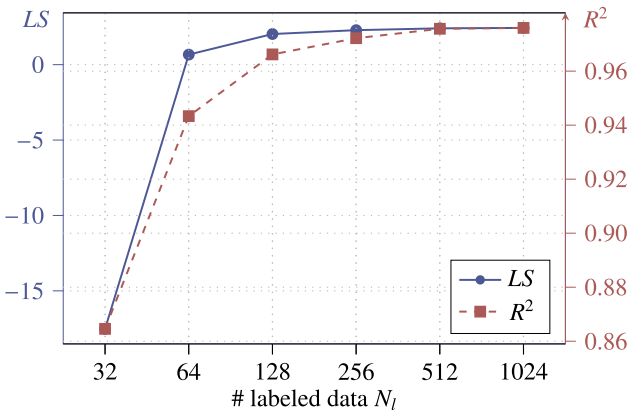


Fig. 7. Predictive performance in terms of the R^2 and LS metrics as a function of the number of labeled data points N_l ($N_u = N_{\mathcal{O}} = 0$), for $d_f = 32$ and $l_\lambda = 0.15$. Results have been averaged by repeatedly training the model on resampled data.

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.

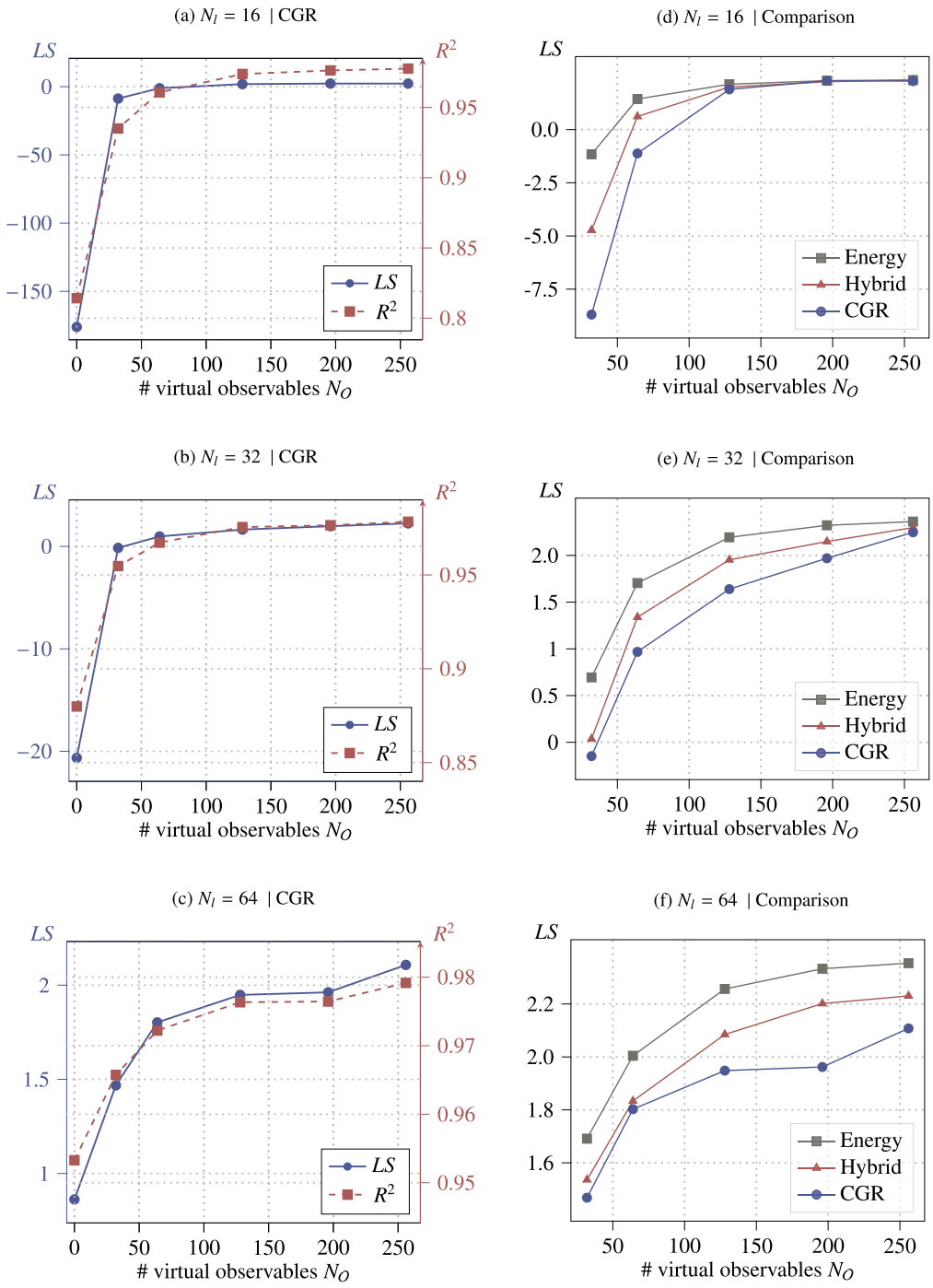


Fig. 8. LEFT COLUMN: Predictive performance of a model trained on N_l labeled data, $N_{\mathcal{O}}$ virtual observables of type CGR ($N_u = 0$). RIGHT COLUMN: Comparison of predictive performance in terms of the *LS* metric with respect to 3 different *types* of virtual observables. The baseline performance 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.

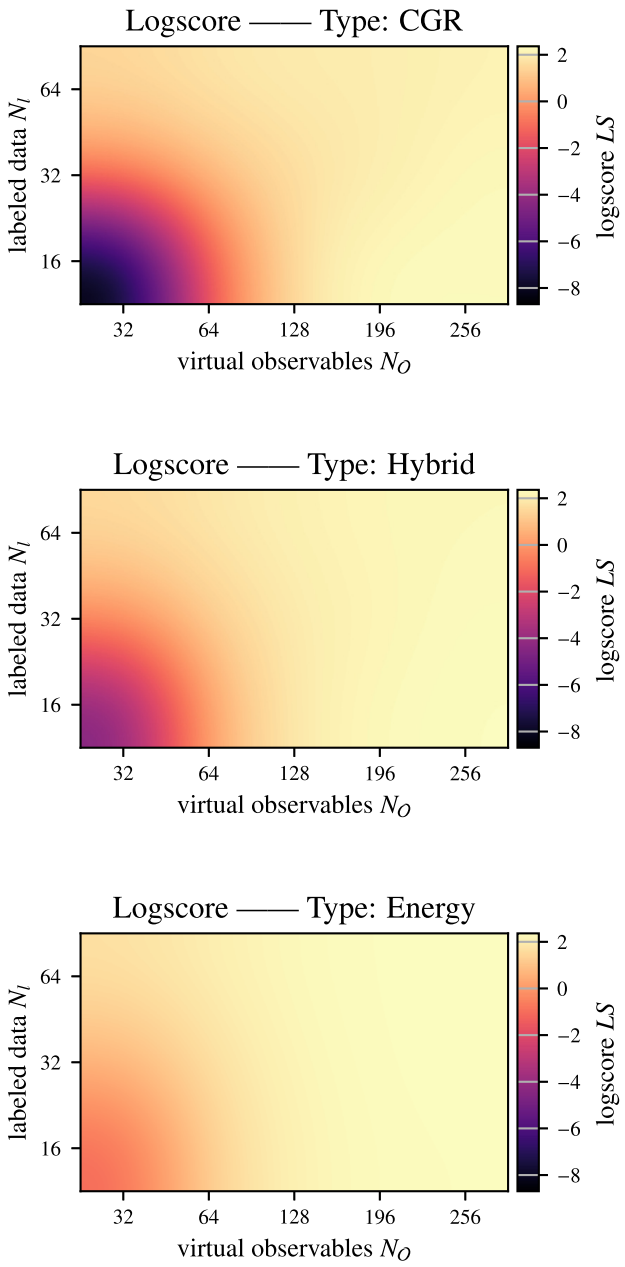


Fig. 9. LS score as function of N_l (number of labeled data) and $N_{\mathcal{O}}$ (number of virtual observables). Results have been averaged by repeatedly training the model on resampled data.

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

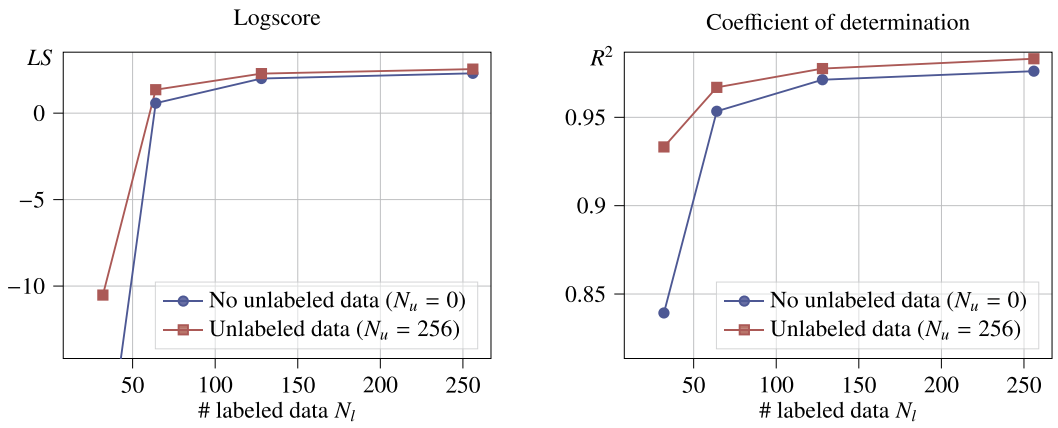


Fig. 10. A model trained on a certain number of labeled data N_l is compared to a model which in addition had access to $N_u = 256$ unlabeled data points, the latter achieving consistently better performance. Results have been averaged by repeatedly training the model on resampled data.

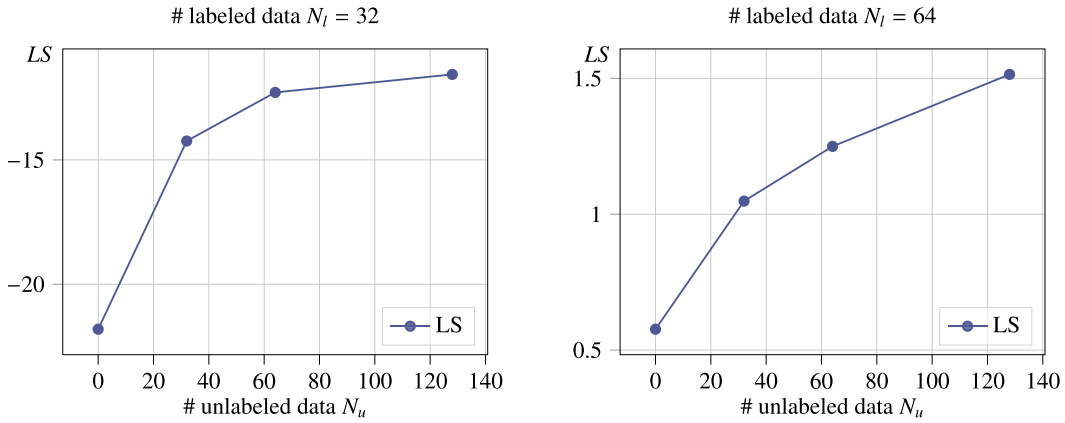


Fig. 11. The predictive performance of the generative model as a function of the number of unlabeled data N_u for $N_l = 32$ (left) and $N_l = 64$ (right). Results have been averaged by repeatedly training the model on resampled data.

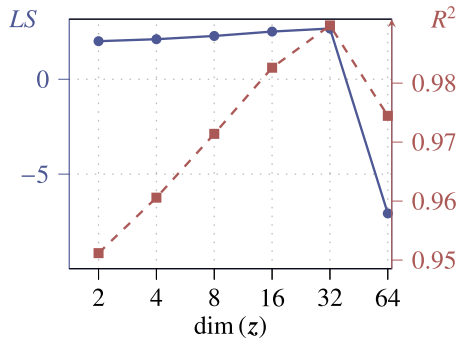
the output $\boldsymbol{y}$ through $\boldsymbol{X}$ i.e. the CGM (Fig. 2). As one can also observe, the benefit of unlabeled data decreases the higher N_l (i.e. the number of labeled data) is. This is not unexpected as the room for improvement is smaller for higher N_l.

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).

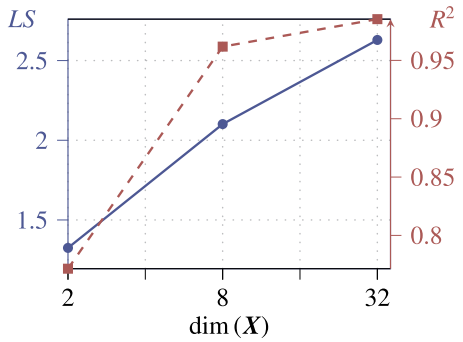
3.7. Effect of the lower-dimensional encoding and the CGM

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

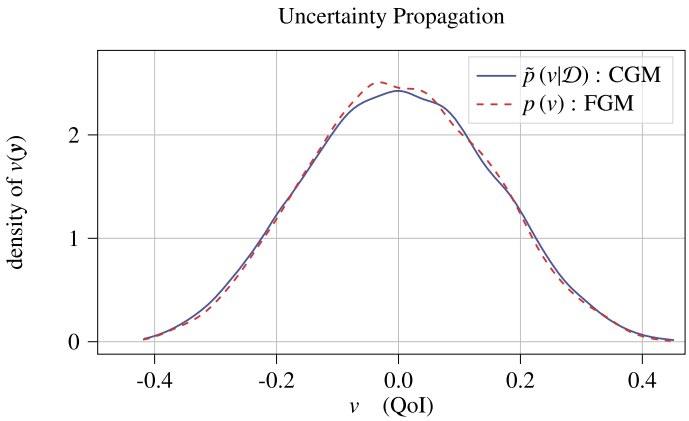


Fig. 13. The predictive posterior density $p(v|\mathcal{D})$ over the QoI $v(\boldsymbol{y})$ as compared with the Monte Carlo reference $p(v)$ obtained with $N_{MC} = 8192$ FGM solves. The model has been trained using $N_l = 32$ (compare this with N_{MC}), $N_u = 8192$ and $N_{\mathcal{O}} = 256$ *hybrid* 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 $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ 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

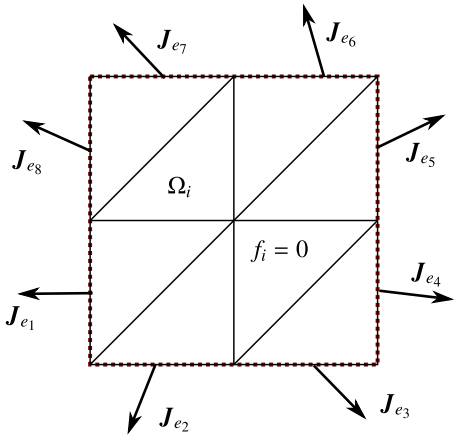


Fig. A.14. If the source term f_i associated with subdomain Ω_i is zero, then the integrated flux across the boundary should net to zero. The discrepancy of this flux $o_i := \Delta\Psi_{\Omega_i}$ corresponds to a virtual observable (equality constraint) introduced as artificial node in our probabilistic 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. $o_i := \Delta\Psi_{\Omega_i} \sim \mathcal{N}\left(0, \tau_i^{-1}\right)$. 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 $q_{\boldsymbol{\xi}}(\boldsymbol{\theta}, \mathcal{R})$ (see Equation (27)), note that the optimal $q^*(\mathcal{Y}_\mathcal{O})$ follows by integrating out all other factors of $q_{\boldsymbol{\xi}}$ [70]

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

where $\hat{\mathcal{O}} = \{\hat{\boldsymbol{o}}\}_{i_\mathcal{O}=1}^{N_\mathcal{O}}$ comprises all virtual observations and $\tilde{q}_{\boldsymbol{\xi}}$ denotes all other factors of the structured mean-field approximation aside from $q(\mathcal{Y}_\mathcal{O})$, i.e. $q_{\boldsymbol{\xi}} = q(\mathcal{Y}_\mathcal{O})\tilde{q}_{\boldsymbol{\xi}}$. 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 $\tilde{q}_{\boldsymbol{\xi}}$. 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 $\boldsymbol{\Gamma}(\boldsymbol{x}^{(i)})^T\boldsymbol{\Lambda}\boldsymbol{\Gamma}(\boldsymbol{x}^{(i)})$ 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 $q(\boldsymbol{y}^{(i_\mathcal{O})}) = \mathcal{N}(\boldsymbol{\mu}^{(i_\mathcal{O})}, \boldsymbol{\Sigma}^{(i_\mathcal{O})})$ at a cost $\mathcal{O}(M^3)$, i.e. numerical expense of updating $q(\boldsymbol{y}^{(i)})$ depends on the number of enforced constraints rather than the dimension of $\boldsymbol{y}$. Making use of the Woodbury matrix identity one finds

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

[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 $\boldsymbol{\Xi}^{(i_\mathcal{O})} = \boldsymbol{\Gamma}(\boldsymbol{x}^{(i_\mathcal{O})})\langle \boldsymbol{S_y} \rangle \boldsymbol{\Gamma}(\boldsymbol{x}^{(i_\mathcal{O})})^T + \boldsymbol{\Lambda}^{-1}$. 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 $i_\mathcal{O} = 1, ..., N_\mathcal{O}$)

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

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 $\mathrm{cond}(\tau \boldsymbol{K}(\boldsymbol{x}^{(i_\mathcal{O})}) + \langle \boldsymbol{S_y}^{-1} \rangle) \leq \mathrm{cond}(\boldsymbol{K}(\boldsymbol{x}^{(i_\mathcal{O})}))$, 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.

C
Paper B

npj | Computational Materials

www.nature.com/npjcompumats

ARTICLE OPEN

Check for updates

Self-supervised optimization of random material microstructures in the small-data regime

Maximilian Rixner[1] and Phaedon-Stelios Koutsourelakis[1,2]✉

While the forward and backward modeling of the process-structure-property chain has received a lot of attention from the materials' community, fewer efforts have taken into consideration uncertainties. Those arise from a multitude of sources and their quantification and integration in the inversion process are essential in meeting the materials design objectives. The first contribution of this paper is a flexible, fully probabilistic formulation of materials' optimization problems that accounts for the uncertainty in the process-structure and structure-property linkages and enables the identification of optimal, high-dimensional, process parameters. We employ a probabilistic, data-driven surrogate for the structure-property link which expedites computations and enables handling of non-differential objectives. We couple this with a problem-tailored active learning strategy, i.e., a self-supervised selection of training data, which significantly improves accuracy while reducing the number of expensive model simulations. We demonstrate its efficacy in optimizing the mechanical and thermal properties of two-phase, random media but envision that its applicability encompasses a wide variety of microstructure-sensitive design problems.

npj Computational Materials (2022)8:46 ; https://doi.org/10.1038/s41524-022-00718-6

INTRODUCTION

Inverting the process-structure-property (PSP) relationships represents a grand challenge in materials science as it holds the potential of expediting the development of new materials with superior performance[1,2]. While significant progress has been made in the forward and backward modeling of the process-structure and structure-property linkages and in capturing the nonlinear and multiscale processes involved[3], much fewer efforts have attempted to integrate uncertainties which are an indispensable component of materials' analysis and design[4,5]. Uncertainties can arise since: (a) process variables do not fully determine the resulting microstructure but rather a probability distribution on microstructures[6], (b) noise and incompleteness are characteristic of experimental data that are used to capture process-structure (most often) and structure-property relations[7], (c) models employed for the process-structure or structure-property links are often stochastic and there is uncertainty in their parameters or form, especially in multiscale formulations[8], and (d) model compression and dimension reduction employed in order to gain efficiency unavoidably lead to some loss of information which in turn gives rise to predictive uncertainty[9]. This randomness should be incorporated, not only in the forward modeling of the PSP chain, but in the optimization objectives and the inverse-design tasks as well.

(Back-)propagating uncertainty through complex and potentially multiscale models poses significant computational difficulties[10]. Data-based surrogates can alleviate these as long as the number of training data, i.e., the number of solutions of the complex models they would substitute, is kept small. In this small-data setting additional uncertainty arises due to the predictive inaccuracy of the surrogate. Quantifying it can not only lead to more accurate estimates but also guide the acquisition of additional experimental/simulation data.

We note that problem formulations based on Bayesian Optimization[11–13] account for uncertainty in the objective solely due to the imprecision of the surrogate and not due to the aleatoric, stochastic variability of the underlying microstructure. In the context of optimization/design problems in particular, a globally-accurate surrogate would be redundant. It would suffice to have a surrogate that can reliably drive the optimization process to the vicinity of the optimum (or optima) and can sufficiently resolve this (those) in order to identify the optimal control parameters. Since the location of the optima is, a priori, unknown, adaptive strategies, in which the training of the surrogate and the optimization are coupled, would be necessary.

We emphasize that unlike successful efforts e.g., in topology optimization[14] or general heterogeneous media[15] which find a single, optimal microstructure maximizing some property-based objective, our goal is more ambitious but also more consistent with the physical reality. We attempt to find the value of the processing variables that gives rise to the optimal *distribution* of microstructures (Fig. 1). To address the computational problem arising from the presence of uncertainties, we recast the stochastic optimization as a probabilistic inference task and employ approximate inference techniques based on Stochastic Variational Inference (SVI[16]).

In terms of the stochastic formulation of the problem, our work most closely resembles that of[17] where they seek to identify a probability density on microstructural features which would yield a target probability density on the corresponding properties. While this poses a challenging optimization problem, producing a probability density on microstructural features does not provide unambiguous design guidelines. In contrast, we operate on (and average over) the whole distribution of microstructures and consider a much wider range of design objectives. In[18] random microstructures were employed but their macroscopic properties were insensitive to their random variability (due to scale-separation) and low-dimensional parametrizations of the two-point correlation function were optimized using gradient-free tools. In a similar fashion, in[19,20] analytic, linear models were

[1]Professorship of Data-driven Materials Modeling, Department of Engineering Physics and Computation, Technical University of Munich, Boltzmannstr. 15, D-85748 Garching bei München, Germany. [2]Munich Data Science Institute (MDSI - Core Member), Garching bei München, Germany. ✉email: p.s.koutsourelakis@tum.de

Published in partnership with the Shanghai Institute of Ceramics of the Chinese Academy of Sciences

npj nature partner journals

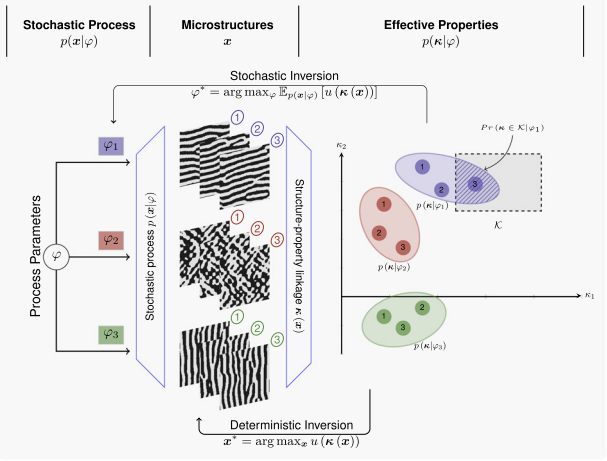


Fig. 1 Conceptual overview. Given stochastic process-structure and structure-property links, we identify the process parameters φ^* which maximize the expected utility $\mathbb{E}_{p(\boldsymbol{x}|\varphi)}[u(\boldsymbol{\kappa})]$ (Illustration based on the specific case $u(\boldsymbol{\kappa}) = \mathbb{I}_{\mathcal{K}}(\boldsymbol{\kappa})$ and $p(\boldsymbol{\kappa}|\boldsymbol{x}) = \delta(\boldsymbol{\kappa} - \boldsymbol{\kappa}(\boldsymbol{x}))$). (Micro) Structures $\boldsymbol{x}$ arise from a stochastic process through the density $p(\boldsymbol{x}|\varphi)$ which depends on the process parameters φ. A data-driven surrogate is employed to predict properties $\boldsymbol{\kappa}$ which introduces additional uncertainty.

employed which, given small and Gaussian uncertainties on the macroscopic properties, find the underlying orientation distribution function (ODF) of the crystalline microstructure. In[21,22], averaged macroscopic properties (ignoring the effects of crystal size and shape) were computed with respect to the ODF of the polycrystalline microstructure and on the basis of their targeted values, the corresponding ODF is found. While data-based surrogates were also employed, the problem formulation did not attempt to quantify the effect of microstructural uncertainties.

In terms of surrogate development, in this work we focus on the microstructure-property link and consider random, binary microstructures, the distribution of which depends on some processing-related parameters. We develop active learning strategies that are tailored to the optimization objectives. The latter can account for the potential stochasticity of the material properties (as well as the predictive uncertainty of the surrogate), i.e., we enable the solution of optimization-under-uncertainty problems.

RESULTS & DISCUSSION

It is advisable that the readers familiarize themselves with the mathematical entities defined in the "Methods" section in order to better appreciate the results presented in this section which contains two applications of the methodological framework, for (O1)- and (O2)-type formulations of the inversion of the PSP chain (see "Methods"). We first elaborate on the specific choices for the process parameters φ, the random microstructures $\boldsymbol{x}$ and their properties $\boldsymbol{\kappa}$ as well as the associated PSP links.

Process φ - Microstructure $\boldsymbol{x}$

In all numerical illustrations we consider statistically homogeneous, binary (two-phase) microstructures which upon spatial discretization (on a uniform, two-dimensional $N_p \times N_p$ grid with $N_p = 64$) are represented by a vector $\boldsymbol{x} \in \{0, 1\}^{4096}$. The binary microstructures are modeled by means of a thresholded zero-mean, unit-variance Gaussian field[23,24]. If the vector $\boldsymbol{x}_g$ denotes the discretized version of the latter (on the same grid), then the value at each pixel i is given by $x_i = H(x_{g,i} - x_0)$ where $H(\cdot)$ denotes the Heaviside function and x_0 the cutoff threshold, which determines the volume fractions of the resulting binary field. We parameterize the spectral density function (SDF) of the underlying Gaussian field (i.e., the Fourier transform of its autocovariance) with φ, using a combination of radial basis functions (RBFs—see Supplementary Notes) which automatically ensures the non-negativity of the resulting SDF. The constraint of unit variance is enforced using a softmax transformation. The density $p(\boldsymbol{x}|\varphi)$ implicitly defined above affords great flexibility in the resulting binary microstructures (as can be seen in the ensuing illustrations) which increases as the dimension of φ does. Figure 1 illustrates how different values of the process parameters φ can lead to profound changes in the microstructures (and correspondingly, their effective physical properties $\boldsymbol{\kappa}$). While the parameters φ selected do not have explicit physical meaning, they can be linked to actual processing variables given appropriate data. Naturally, not all binary media can be represented by this model and a more flexible $p(\boldsymbol{x}|\varphi)$, potentially learned from actual process-structure data, could be employed with small modifications in the overall algorithm[25–27].

Microstructure $\boldsymbol{x}$ - Properties $\boldsymbol{\kappa}$

In this study we consider a two-dimensional, representative volume element (RVE) $\Omega_{\text{RVE}} = [0, 1]^2$ and assume each of the two phases are isotropic, linear elastic in terms of their mechanical response and are characterized by isotropic, linear conductivity tensors in terms of their thermal response. We denote with $\mathbb{C}$ the fourth-order elasticity tensor and with $\mathbf{a}$ the second order conductivity tensor which are also binary (tensor) fields. The vector $\boldsymbol{\kappa}$ consists of various combinations of macroscopic, effective (apparent), mechanical or thermal properties of the RVE which we denote by $\mathbb{C}^{\text{eff}}$ and $\mathbf{a}^{\text{eff}}$, respectively. The effective properties for each microstructure occupying Ω_{RVE} were computed using finite element simulations and Hill's averaging theorem[28,29] (further details are provided in the Supplementary Notes). We assumed a contrast ratio of 50 in the properties of the two phases, i.e., $E_1/E_0 = 50$ (where E_0, E_1 are the elastic moduli of phases 0 and 1, as well as Poisson's ratio $\nu = 0.3$ for both phases) and $a_1/a_0 = 50$ (where a_0, a_1 are the conductivities of phases 0 and 1). In the following plots, phase 1 is always shown with white and phase 0 with black. We note that the dependence of effective properties on (low-dimensional) microstructural features (analogous to φ) has been considered, in e.g.[30,31], but the random variability in these properties has been ignored either by considering very large RVEs or by averaging over several of them. We emphasize finally that the framework proposed can accommodate any high-fidelity model for the structure-property link as this is merely used as a generator for the training data $\mathcal{D}$.

Case 1: Target domain of multi-physics properties (O1). In the following we will demonstrate the performance of the proposed formulation in an (O1)-type stochastic optimization problem (see "Methods"), with regards to both thermal as well as mechanical properties. In addition, we will provide a systematic and quantitative assessment of the benefits of the active learning strategy proposed (as compared to randomized data generation).

We consider a combination of mechanical and thermal properties of interest, namely [Eq. 1]:

$$\kappa_1 = [\boldsymbol{a}^{\text{eff}}]_{11}, \qquad \kappa_2 = \frac{1}{2}\left([\mathbb{C}^{\text{eff}}]_{1111} + [\mathbb{C}^{\text{eff}}]_{2222}\right) \tag{1}$$

i.e., $\boldsymbol{\kappa} \in \mathbb{R}^2_+$, and define the target domain [Eq. 2]:

$$\mathcal{K} = [8.5,\ 11.0] \times [6.75,\ 9.0] \tag{2}$$

The utility function $u(\boldsymbol{\kappa}) = \mathbb{I}_{\mathcal{K}}(\boldsymbol{\kappa})$ is the (non-differentiable) indicator function of $\mathcal{K} \subset \mathbb{R}^2_+$ which implies that the objective of the optimization (type (O1)—see Fig. 2a) is to *find the φ that maximizes the probability that the resulting microstructures have properties $\boldsymbol{\kappa}$ that lie in $\mathcal{K}$*. The two-phase microstructures have

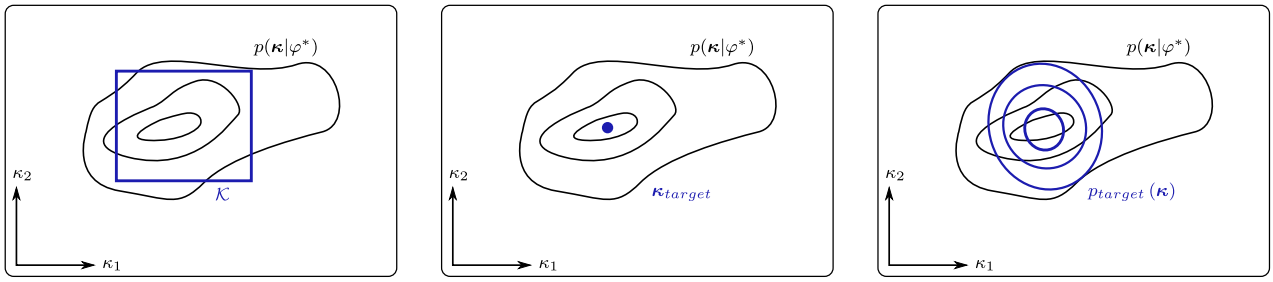
volume fraction 0.5 and the parameters $\varphi \in \mathbb{R}^{100}$ as well as $p(\boldsymbol{x}|\varphi)$ were defined as discussed in the beginning of this section.

With regards to the adaptive learning strategy (appearing as the outer loop in Algorithm (1) in "Methods"), we note that the initial training dataset $\mathcal{D}^{(0)}$ consists of $N_0 = 2048$ data pairs which are generated via ancestral sampling, i.e., we randomly draw samples φ from $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ and conditionally on each $\varphi^{(n)}$ we sample $p(\boldsymbol{x}|\varphi^{(n)})$ to generate a microstructure (the choice $\varphi \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ is not arbitrary, as—given the adopted parametrization—it envelopes all possible SDFs). In each data acquisition step l, $N_{pool} = 4096$ candidates were generated and a subset of $N_{add} = 1024$ of those was selected based on the acquisition function. We note that N_{add} (as well as N_0) defines a trade-off between information acquisition and computational cost. Hence the size of the dataset increased by 1024 data pairs at each iteration l, with $L = 4$ data augmentation steps performed in total.

The optimal process parameters at each data acquisition step are denoted as $\varphi^*_{\mathcal{M},\mathcal{D}^{(l)}}$, with the subscript indicating the dependence on the surrogate model $\mathcal{M}$ and the dataset $\mathcal{D}^{(l)}$ on which it has been trained. Once the algorithm has converged to its final estimate of the process parameters after L data acquisition steps, i.e., $\varphi^*_{\mathcal{M},\mathcal{D}^{(L)}}$, we can assess $\varphi^*_{\mathcal{M},\mathcal{D}^{(L)}}$ by obtaining a *reference* estimate of the expected utility $U(\varphi^*_{\mathcal{M},\mathcal{D}^{(L)}}) = Pr(\boldsymbol{\kappa} \in \mathcal{K}|\varphi^*_{\mathcal{M},\mathcal{D}^{(L)}})$ using Monte Carlo, i.e., by sampling microstructures $\boldsymbol{x} \sim p(\boldsymbol{x}|\varphi^*_{\mathcal{M},\mathcal{D}^{(L)}})$, and running the high-fidelity model instead of the inexpensive surrogate. In this manner we can also compare the optimization results obtained with active learning with those obtained by using randomized training data $\mathcal{D}$ (i.e., without adaptive learning). We argue that the former has a competitive advantage, if for the same total number N of datapoints we can achieve a higher score in terms of our materials' design objective $Pr(\boldsymbol{\kappa} \in \mathcal{K}|\varphi^*)$. As the optimization objective $\mathcal{F}$ is non-convex and the optimization algorithm itself non-deterministic, generally the optimal process parameters φ^* identified can vary across different runs (non-determinancy arises from the randomized generation of the data, the stochastic initialization of the neural network, as well as the randomized initial guess of $\varphi^{(0)} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$) . For this reason the optimization problem is solved several times (with different randomized initializations) and we report on the aggregate performance of active learning vs. randomized data generation (baseline).

In the following we discuss the results obtained and displayed in Figs. 3, 4, 5 and 6.

- In Fig. 3 we depict sample microstructures drawn from $p(\boldsymbol{x}|\varphi)$ for two values of φ, i.e., for the initial guess $\varphi^{(0)}$ (Fig. 3a) and for optimal process parameters $\varphi^*_{\mathcal{M},\mathcal{D}^{(L)}}$ (Fig. 3b). While the optimized microstructures as shown in Fig. 3b remain random, one observes that the connectivity of phase 1 (stiffer) is increased as compared to the microstructures shown in Fig. 3a. The diagonal, connected paths of the lesser



(a) Target domain $\mathcal{K}$ **(b)** Target value $\boldsymbol{\kappa}_{target}$ **(c)** Target distribution $p_{target}(\boldsymbol{\kappa})$

Fig. 2 Illustration of various materials design objectives. Different optimization objectives with respect to the density $p(\boldsymbol{\kappa}|\varphi)$ that expresses the likelihood of property values $\boldsymbol{\kappa}$ for given processing conditions φ. We illustrate the following cases: (**a**) we seek to maximize the probability that the material properties $\boldsymbol{\kappa}$ fall within a target domain $\mathcal{K}$. (**b**) We seek to minimize the mean deviation of the properties $\boldsymbol{\kappa}$ from a target value $\boldsymbol{\kappa}_{target}$. (**c**) we seek to minimize the deviation between $p(\boldsymbol{\kappa}|\varphi)$ and a target probability density $p_{target}(\boldsymbol{\kappa})$ on the material properties.

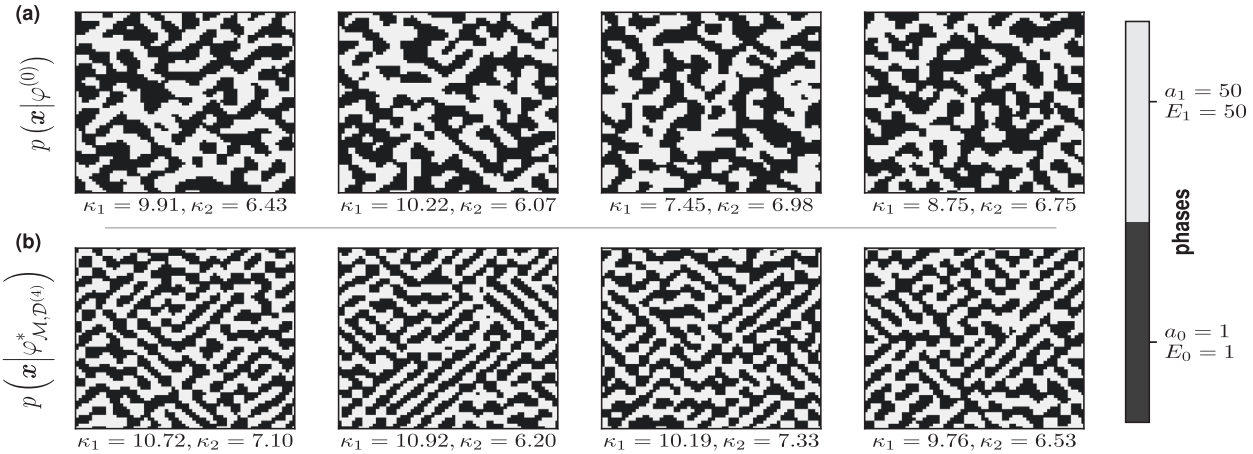


Fig. 3 Case 1: Optimal random microstructures. (**a**) Samples of microstructures drawn from $p(\boldsymbol{x}|\varphi)$ for the initial guess $\varphi^{(0)}$ of processing variables. (**b**) Samples of microstructures drawn from $p(\boldsymbol{x}|\varphi)$ for the optimal value $\varphi^*_{\mathcal{M},\mathcal{D}^{(L)}}$ of processing variables which maximize the probability that the corresponding material properties will fall in the target domain $\mathcal{K} = [8.5, 11.0] \times [6.75, 9.0]$ (Eq. (2)). Underneath each microstructure, the thermal κ_1 and mechanical κ_2 properties of interest (Eq. (1)) are reported.

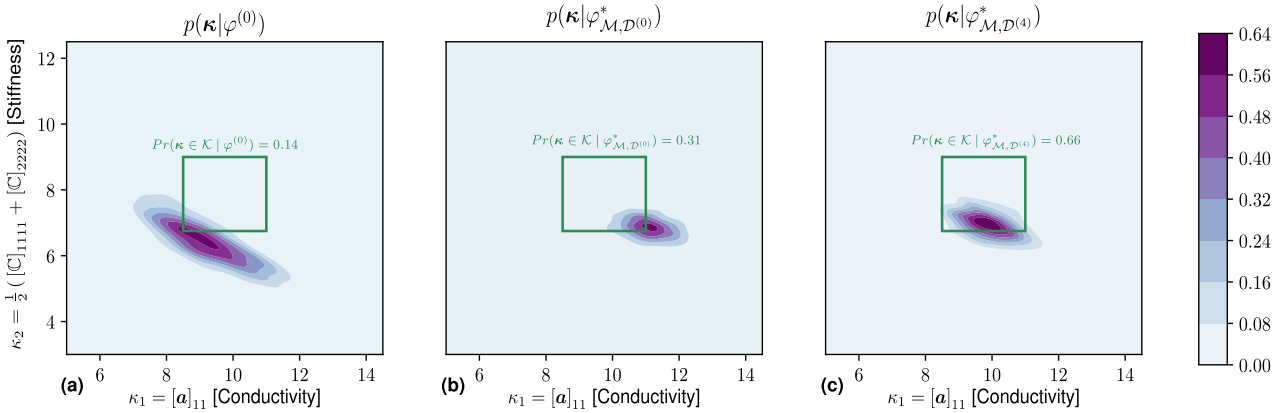


Fig. 4 Case 1: Evolution of the process-property density during optimization. The actual process-property density $p(\boldsymbol{\kappa}|\varphi)$ was estimated using 1024 Monte Carlo samples making use of the high-fidelity structure-property model (see Supplementary Notes), and for the following three values of the process parameters φ: (**a**) for the initial guess $\varphi^{(0)}$, (**b**) for the optimal φ as obtained using the initial training dataset $\mathcal{D}^{(0)}$ and without adaptive learning, (**c**) for the optimal φ obtained with the augmented training dataset $\mathcal{D}^{(4)}$ identified by the active learning scheme proposed. The target domain $\mathcal{K}$ (Eq. (2)) is drawn with a green rectangle and the colorbar indicates the value of the density $p(\boldsymbol{\kappa}|\varphi)$.

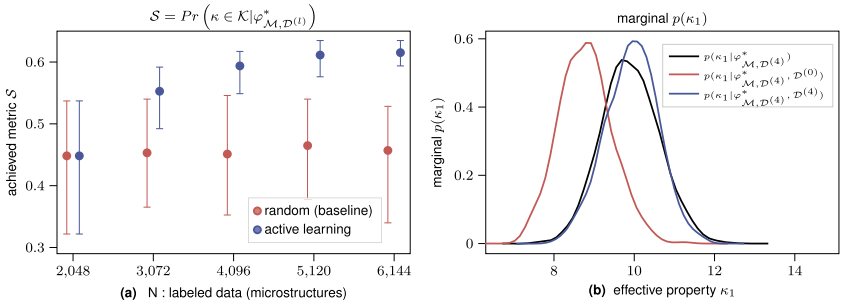


Fig. 5 Case 1: Assessment of active learning approach. (**a**) The probability we seek to maximize with respect to φ, i.e., $Pr(\boldsymbol{\kappa} \in \mathcal{K}|\varphi)$ is plotted as a function of the size N of the training dataset (i.e., the number of simulations of the high-fidelity model). Based on 80 independent runs of the optimization algorithm, we plot the median value (with dots) and the 50% probability quantiles (with error bars). The red lines correspond to the results obtained without adaptive learning and the blue with adaptive learning. (**b**) For the the optimal $\varphi^{*}_{\mathcal{M},\mathcal{D}^{(4)}}$ identified using active learning, we compare the actual process-property density $p(\kappa_1|\varphi)$ (black line—estimated with 1024 Monte Carlo samples and the high-fidelity model) with the one predicted by the surrogate trained only on the initial dataset $\mathcal{D}^{(0)}$ (red line) and with the one predicted by the surrogate trained on the augmented dataset $\mathcal{D}^{(4)}$ (blue line).

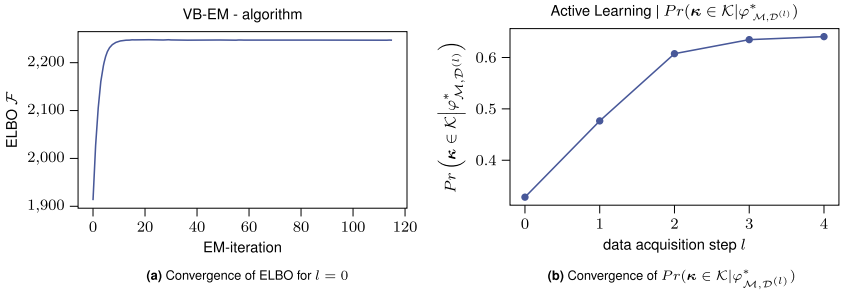


Fig. 6 Case 1: Convergence characteristics of the optimization algorithm. We illustrate for a single optimization run (**a**) the evolution of the ELBO $\mathcal{F}$ as a function of the iteration number in the inner loop and for $l = 0$ (outer loop—see Algorithm (1)). (**b**) Evolution of the probability we seek to maximize $Pr(\boldsymbol{\kappa} \in \mathcal{K}|\varphi)$ (estimated with 1024 Monte Carlo samples and the high-fidelity model) for the optimal values $\varphi^{*}_{\mathcal{M},\mathcal{D}^{(l)}}$ identified by the algorithm at various data acquisition steps l (outer loop in Algorithm (1)).

conducting phase (black) effectively block heat conduction in the horizontal direction. This is also reflected in the effective properties reported underneath each image. The value of the objective, i.e., the probability that properties $\boldsymbol{\kappa}$ reside in $\mathcal{K}$, is ≈ 0.65 for the optimal microstructures (Fig. 3b), as opposed to ≈ 0.14 for the microstructures shown in Fig. 3b (see also Fig. 4a and c).

- Figure 4 provides insight into the optimization algorithm proposed by looking at the process-property density $p(\boldsymbol{\kappa}|\varphi)$ for various φ values. We note that this density is implicitly defined by propagating the randomness in the microstructures (quantified by $p(\boldsymbol{x}|\varphi)$) through the high-fidelity model that predicts the properties of interest. Based on the Monte Carlo estimates depicted in Fig. 4, one observes that the density

$p(\boldsymbol{\kappa}|\varphi)$ only minimally touches the target domain $\mathcal{K}$ for initial process parameters $\varphi^{(0)}$ (Fig. 4a) and gradually moves closer to $\mathcal{K}$ as the iterations proceed, with the optimization informed by the surrogate trained on the initial batch of data $\mathcal{D}^{(0)}$ (Fig. 4b). The incorporation of additional training data by means of the adaptive learning scheme enables the surrogate to resolve the details in the structure-property map with sufficient detail to eventually identify process parameters such that the density $p(\boldsymbol{\kappa}|\varphi)$ maximally overlaps (in comparison) with the target domain $\mathcal{K}$ (Fig. 4c).

- In Fig. 5a we illustrate the performance advantage gained by the active learning approach proposed over the baseline. To this end, we compare the values of the objective function, i.e., $Pr(\boldsymbol{\kappa} \in \mathcal{K}|\varphi^*_{\mathcal{M},\mathcal{D}})$ achieved for datasets $\mathcal{D}$ of equal size, with the dataset being either generated randomly (baseline), or constructed based on our active learning approach. Evidently, the latter was able to achieve a better material design at comparably significantly lower numerical cost (as measured by the number of evaluations of the high-fidelity model of the S-P link). We observe that while the addition of more training data generally leads to more accurate surrogates, when this is done without regard to the optimization objectives (red line), then it does not necessarily lead to higher values of the objective function. In Fig. 5b we provide further insight as to why the adaptive data acquisition was able to outperform a randomized approach. To this end we consider the impact of adaptive learning on the model belief for one of the effective properties $\boldsymbol{\kappa}$, i.e., we compare the model-based belief $p(\kappa_1|\varphi^*_{\mathcal{M},\mathcal{D}^{(4)}}, \mathcal{D}^{(0)})$ of the surrogate conditional on $\mathcal{D}^{(0)}$ against a reference density obtained using Monte Carlo (black line). We can see that a model only informed by $\mathcal{D}^{(0)}$ (red line) identifies an incorrect density and as such fails to converge to the optimal process parameters. The active learning approach (blue line) was able to correct the initially erroneous model belief and as a result performs better in the optimization task.
- In Fig. 6a we illustrate the evolution of the ELBO during the inner-loop iterations of the proposed VB-EM algorithm (see Algorithm (1) in "Methods"). Finally, in Fig. 6b we depict the evolution of the maximum of the objective identified at various data acquisition steps l of the proposed active learning scheme in a single, indicative run (in contrast to Fig. 5 where results over multiple runs are summarized). As it can be seen, the targeted data enrichment enables the surrogate to resolve details in the structure-property map and identify higher-performing processing parameters φ.

Case 2: Target density of properties (O2). In this second numerical illustration, we investigate the performance of the proposed methodological framework for an (O2)-type optimization problem (Eq. (7)) where we seek to *identify the processing parameters φ that lead to a property density $p(\boldsymbol{\kappa}|\varphi)$ that is closest to a prescribed target $p_{target}(\boldsymbol{\kappa})$*. In particular, we considered the following two properties [Eq. 3]

$$\kappa_1 = \left[\boldsymbol{a}^{\text{eff}}\right]_{11} \qquad \kappa_2 = \left[\boldsymbol{a}^{\text{eff}}\right]_{22} \tag{3}$$

i.e., $\boldsymbol{\kappa} \in \mathbb{R}^2$ and a target density [Eq. 4]:

$$p_{target}(\boldsymbol{\kappa}) = \mathcal{N}(\hat{\boldsymbol{\mu}}, \hat{\Sigma}) \tag{4}$$

with $\hat{\boldsymbol{\mu}} = [20.5, 3.5]^T$ and $\hat{\Sigma}_{11} = 0.60$, $\hat{\Sigma}_{22} = 0.01, \hat{\Sigma}_{12} = -0.03$ (depicted with green iso-probability lines in Fig. 8). These values were selected to promote anisotropic behavior, i.e., the targeted microstructures should have a large effective conductivity in the first spatial dimension and simultaneously be (relatively) insulating in the second spatial dimension. The characteristics of the active learning procedure (outer loop in Algorithm (1) in "Methods") remain identical, with the only difference that $\mathcal{D}^{(0)}$ now comprises $N_0 = 4096$ datapoints, with $N_{add} = 1024$ datapoints (out of 4096 candidates) added in each of the $L = 6$ data-enrichment steps. We used $S = 20$ samples from $p_{target}(\boldsymbol{\kappa})$ to approximate the objective (see Eq. (9)).

We discuss the results obtained based on Figs. 7 and 8:

- In Fig. 7 we showcase sample microstructures drawn from $p(\boldsymbol{x}|\varphi)$ both for the initial guess $\varphi^{(0)}$ (Fig. 7a, b) as well as for the optimal process parameters $\varphi^*_{\mathcal{M},\mathcal{D}^{(L)}}$ identified by the optimization algorithm using the active learning approach (Fig. 7c, d). The examples shown in Fig. 7a, b correspond to volume fraction 0.5 whereas the examples shown in Fig. 7b, d correspond to volume fraction 0.3 (of the more conducting, white phase, a_1) . As one would expect, we observe that the optimal family of microstructures identified (determined by $\varphi^*_{\mathcal{M},\mathcal{D}^{(L)}}$) exhibit connected paths of the more conductive phase (white) along the horizontal direction. The connected paths of the lesser conducting phase (black) are also aligned

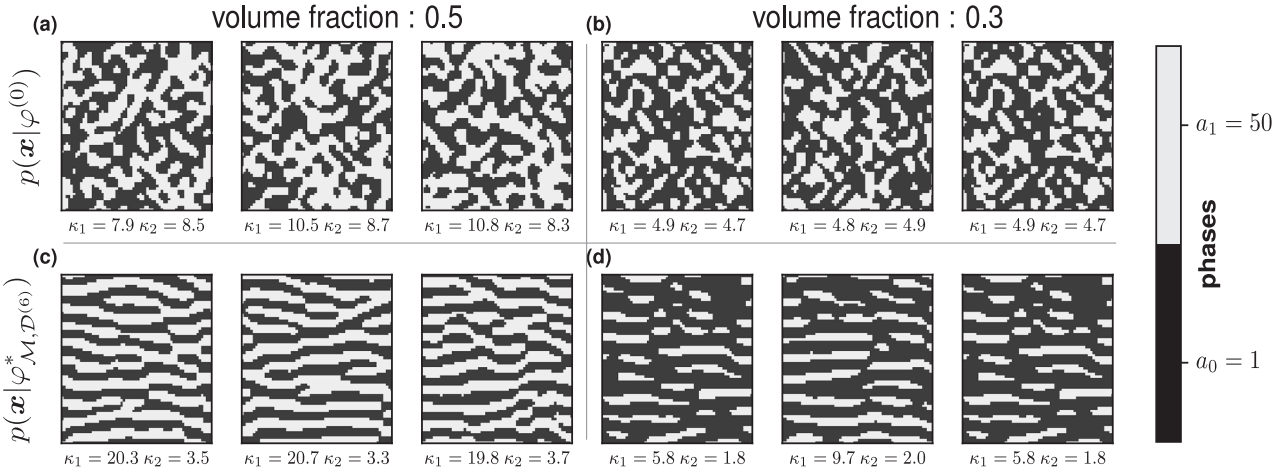


Fig. 7 Case 2: Optimal random microstructures. (**a**), (**b**) Samples of microstructures drawn from $p(\boldsymbol{x}|\varphi)$ for the initial guess $\varphi^{(0)}$ of processing variables. (**c**), (**d**) Samples of microstructures drawn from $p(\boldsymbol{\kappa}|\varphi)$ for the optimal value $\varphi^*_{\mathcal{M},\mathcal{D}^{(6)}}$ of the processing variables which minimize the KL-divergence between $p(\boldsymbol{\kappa}|\varphi)$ and the target density $p_{target}(\boldsymbol{\kappa})$ (Eq. (4)). Underneath each microstructure, the thermal properties κ_1, κ_2 of interest (Eq. (3)) are reported. The illustrations correspond to two volume fractions 0.5 (in (**a**, **c**)) and 0.3 (in (**b**, **d**)) of the high-conductivity phase ($a_1 = 50$).

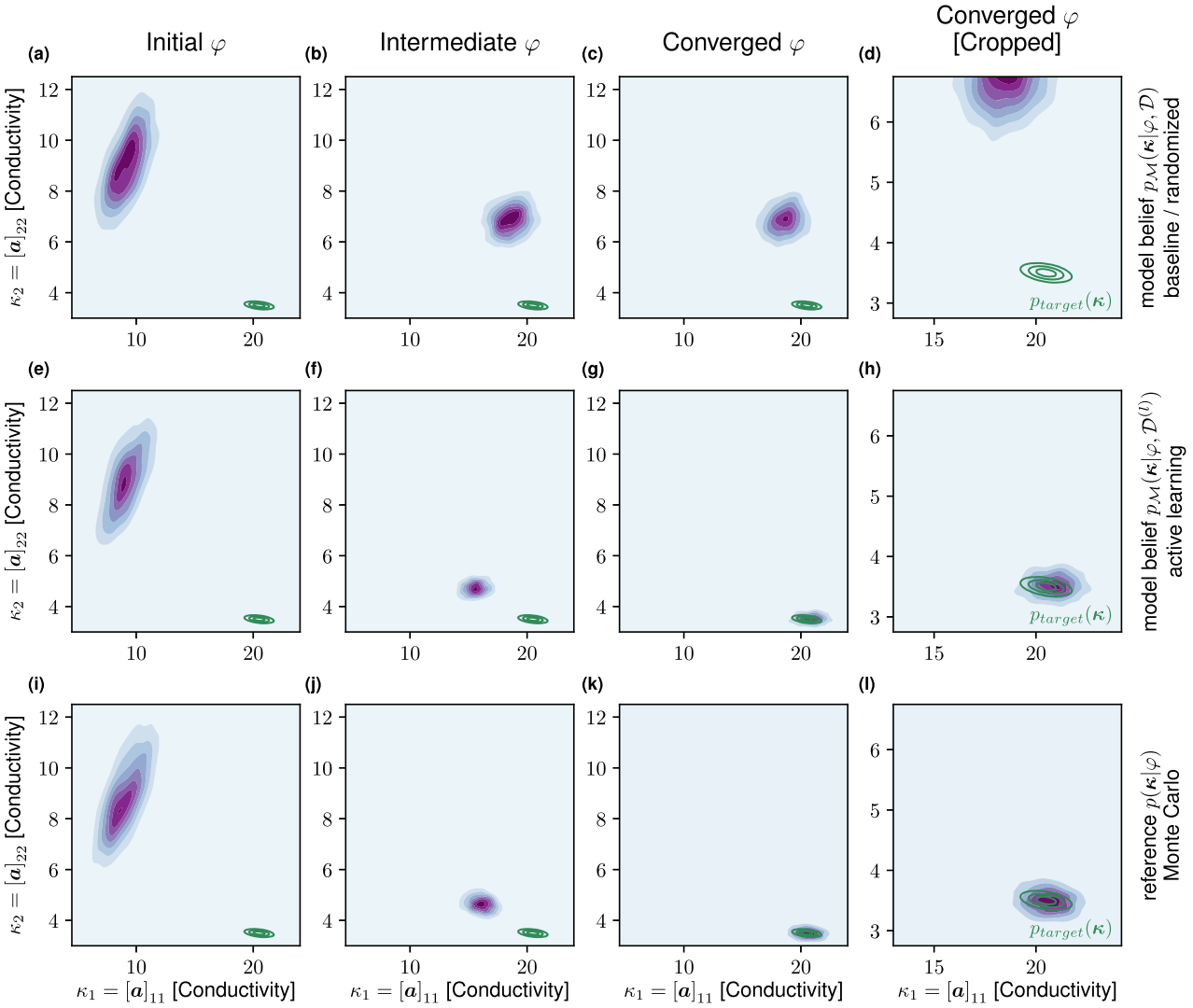


Fig. 8 Case 2: Evolution the of process-property density $p(\kappa|\varphi)$ with and without active learning in relation to the target $p_{target}(\kappa)$. We plot the evolution of the process-property density $p(\kappa|\varphi)$ at three different stages of each optimization run, i.e.: the initial φ (**a**, **e**, **i**), the φ at an intermediate stage of the optimization (**b**, **f**, **j**), and the optimal φ identified upon convergence (**c**, **g**, **k**). The fourth column, i.e., (**d**, **h**, **l**) is a zoomed-in version of the third that enables closer comparisons of the densities involved. (**a–d**) Illustrate $p(\kappa|\varphi)$ as predicted by the surrogate trained on a randomized dataset without active learning. (**e–h**) Illustrate $p(\kappa|\varphi)$ as predicted by the surrogate trained using the adaptive learning proposed. (**i–l**) Illustrate the actual $p(\kappa|\varphi)$ (estimated with 1024 Monte Carlo samples and the high-fidelity model) and for the optimal φ identified by the active learning approach. The target distribution $p_{target}(\kappa)$ is indicated with green iso-probability lines.

in the horizontal direction so as to reduce the effective conductivity along the vertical direction. The optimal microstructures therefore exhibit a marked anistropy and funnel heat through pipe-like structures of high-conductivity material in the horizontal direction. This is also reflected in the indicative property values reported under each frame.

- Finally, Fig. 8 assesses the advantage of the active learning strategy advocated for this problem. In particular, we plot the evolution of the process-structure density $p(\kappa|\varphi)$ in relation to the target $p_{target}(\kappa)$ (depicted with green iso-probability lines) at different stages of the optimization (initial-intermediate-converged). Using the optimal process parameters φ identified at each of these stages, we see that the optimization scheme *without* active learning (Fig. 8a–d) results in a density that is quite far from the target. In contrast, the optimization algorithm *with* active learning (Fig. 8e–h) is able to identify a φ which brings the $p(\kappa|\varphi)$ very close to the target distribution $p_{target}(\kappa)$. The validity of this result is assessed in (Fig. 8i–l) where the actual $p(\kappa|\varphi)$ (estimated with Monte Carlo and the high-fidelity model) is depicted for the φ values identified by the active learning approach (Fig. 8e–h). We observe a very close agreement which reinforces previous evidence on the advantages of the active learning strategy advocated.

In conclusion, we presented a flexible, fully probabilistic, data-driven formulation for materials design that can account for the multitude of uncertainties along the the PSP chain and enables the identification of optimal, high-dimensional, process parameters φ.

The methodology relies on probabilistic models or surrogates for the process-structure $p(\kappa|\varphi))$ and structure-property $p(\kappa|\varphi))$ links which could be learned from experimental or simulation data. Although only the latter was extensively discussed in this work, similar concepts and tools can be employed for the construction of the former. The predictive uncertainty of the surrogate is incorporated in the optimization objectives and the self-supervised, active learning mechanism can reduce the requirements on training data, which is particularly important

when those arise from expensive experiments/simulators, so that only the regions necessary for the solution of the optimization problem are resolved. Adaptations to different material descriptions or underlying physics would only require alterations of these densities.

We have demonstrated that a variety of different objectives can be accommodated by appropriate selection of the utility function. Despite the use of surrogates, the computation of the objective functions and their derivatives remains intractable as it requires expectations with respect to the, generally, very high-dimensional microstructural representations. To this end, we employed an Expectation-Maximization scheme which iteratively identifies a (near)-optimal sampling density for estimating the expectations involved while simultaneously updating the estimates for the optimal processing variables.

While not discussed, it is also possible to assess the optimization error, albeit with additional runs of the high-fidelity model, by using an Importance Sampling step[32]. Lastly we mention further potential for improvement by a fully Bayesian treatment of the surrogate's parameters $\boldsymbol{\theta}$, which would be particularly beneficial in the small-data regime we are operating in.

METHODS

A conceptual overview of the proposed stochastic-inversion framework is provided in Fig. 1 where it is contrasted with deterministic formulations. We present the main building blocks and modeling assumptions and subsequently define the optimization problems of interest. We then discuss associated challenges, algorithmic steps and conclude this section with details regarding the probabilistic surrogate model and the active learning strategy.

We define the following variables/parameters:

- process parameters $\boldsymbol{\varphi} \in \mathbb{R}^{d_\phi}$: These are the optimization variables and can parametrize actual processing conditions (e.g., chemical composition, annealing temperature) or statistical descriptors (e.g., ODF) that might be linked to the processing. The higher the dimension of $\boldsymbol{\varphi}$, the more control one has over material design and the more difficult the problem becomes.
- *random* microstructures $\boldsymbol{x}$: This is in general a very high-dimensional vector that represents the microstructure with the requisite detail to predict its properties. In the numerical illustrations which involve two-phase media in $d = 2$ dimensions represented on a uniform grid with N_p subdivisions per dimension, $\boldsymbol{x} \in \{0,1\}^{N_p^d}$ consists of binary variables which indicate the material phase of each pixel (see, e.g., Fig. 3) (for notational simplicity we nonetheless treat $\boldsymbol{x}$ as continuous in general expressions, i.e., define integrals instead of sums). We emphasize that $\boldsymbol{x}$ is a *random vector* due to the stochastic variability of microstructures even in cases where $\boldsymbol{\varphi}$ is the same (see process-structure link below).
- properties $\boldsymbol{\kappa}$: This vector represents the material properties of interest which depend on the microstructure $\boldsymbol{x}$. We denote this dependence with some abuse of notation as $\boldsymbol{\kappa}(\boldsymbol{x})$ and discuss it in the structure-property link below. Due to this dependence, $\boldsymbol{\kappa} \in \mathbb{R}^{d_k}$ will also be a random vector. In the numerical illustrations $\boldsymbol{\kappa}$ consists of mechanical and thermal, effective (apparent) properties.

Furthermore, our formulation includes the:

- process-structure link: We denote the dependence between $\boldsymbol{\varphi}$ and $\boldsymbol{x}$ with the conditional density $p(\boldsymbol{x}|\boldsymbol{\varphi})$ (Fig. 1), reflecting the fact that processing parameters $\boldsymbol{\varphi}$ do not in general uniquely determine the microstructural details. Formally experimental data[25,33] and/or models[34] would need to be used to determine $p(\boldsymbol{x}|\boldsymbol{\varphi})$,which could induce additional uncertainty (see discussion in the Introduction). We also note that no a-priori dimensionality reduction is implied, i.e., the full microstructural details are retained and used in the property-predicting, high-fidelity models. In this work, we assume the process-structure link $p(\boldsymbol{x}|\boldsymbol{\varphi})$ is given a-priori, and its particular form for the binary media examined is detailed in the Results & Discussion section (the binary microstructures considered for our numerical illustrations could arise from the solution of the Cahn–Hilliard equation describing phase separation occurring in a binary alloy under thermal annealing).
- structure-property link: The calculation of the properties $\boldsymbol{\kappa}$ for a given microstructure $\boldsymbol{x}$ involves in general the solution of a stochastic or deterministic, complex, high-fidelity model (in our numerical illustrations, this consists of partial differential equations). We denote the corresponding conditional density as $p(\boldsymbol{\kappa}|\boldsymbol{\varphi})$, which in the case of a deterministic model degenerates to a Dirac-delta. In order to perform the optimization, repeated solutions of the high-fidelity model would be necessary. In a high-dimensional setting, additionally derivatives of $\boldsymbol{\kappa}$ w.r.t. $\boldsymbol{x}$ would in general be required to drive the search. Such derivatives might be either unavailable (e.g., when $\boldsymbol{x}$ is binary as above), or, at the very least, would add to the overall computational burden. To overcome this major efficiency hurdle we advocate the use of a data-driven surrogate model. We denote with $\mathcal{D}$ the training data (i.e., pairs of inputs-microstructures and outputs-properties $\boldsymbol{\kappa}(\boldsymbol{x})$) and explain in the sequel how these are selected (see section on Active Learning). We employ a *probabilistic* (for reasons we explain in the subsequent sections) surrogate model (see Fig. 9) denoted by $\mathcal{M}$ and use $p_{\mathcal{M}}(\boldsymbol{\kappa}|\boldsymbol{x},\mathcal{D})$ to denote its predictive density.

We note that the introduction of $p(\boldsymbol{\kappa}|\boldsymbol{x})$ and $p(\boldsymbol{x}|\boldsymbol{\varphi})$ as a probabilistic representation of the PSP chain is a very general description which in principle can accommodate any epistemic or aleatoric source of uncertainty. With these definitions in hand, we proceed to define two closely related optimization problems (O1) and (O2) that we would like to address. For the first optimization problem (O1) we make use of a utility function $u(\boldsymbol{\kappa}) \geq 0$ (negative-valued utility functions can also be employed, as long as they are bounded from below). Due to the aforementioned uncertainties we consider the *expected utility* $U_1(\boldsymbol{\varphi})$ which is defined as [Eq. 5]:

$$U_1(\boldsymbol{\varphi}) = \mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{\varphi})}\left[\int u(\boldsymbol{\kappa})\, p(\boldsymbol{\kappa}|\boldsymbol{x})\, d\boldsymbol{\kappa}\right] \tag{5}$$

(where $\mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{\varphi})}[.]$ implies an expectation with respect to $p(\boldsymbol{x}|\boldsymbol{\varphi})$) and seek the processing parameters $\boldsymbol{\varphi}$ that maximize it, i.e. [Eq. 6]:

$$(\mathrm{O1}): \qquad \boldsymbol{\varphi}^* = \arg\max_{\boldsymbol{\varphi}} U_1(\boldsymbol{\varphi}) \tag{6}$$

Consider for example the case that $u(\boldsymbol{\kappa}) = \mathbb{I}_{\mathcal{K}}(\boldsymbol{\kappa})$, i.e., the indicator function of some target domain $\mathcal{K}$, defining the desired range of property values (Fig. 2a). In this case, solving (O1) above will lead to the value of $\boldsymbol{\varphi}$ that *maximizes* the probability that the resulting material will have properties in the target domain $\mathcal{K}$, i.e., $U_1(\boldsymbol{\varphi}) = Pr(\boldsymbol{\kappa} \in \mathcal{K}|\boldsymbol{\varphi})$. Similar *probabilistic* objectives have been proposed for several other materials' classes and models (e.g.[35]). Another possibility of potential practical interest involves introducing $u(\boldsymbol{\kappa}) = e^{-\tau||\boldsymbol{\kappa}-\boldsymbol{\kappa}_{target}||^2}$, with τ a scaling parameter. In this case solving (O1) leads to the material with properties which, on average, are closest to the prescribed target $\boldsymbol{\kappa}_{target}$ (Fig. 2b).

The second problem we consider involves prescribing a target density $p_{target}(\boldsymbol{\kappa})$ on the material properties and seeking the $\boldsymbol{\varphi}$ that leads to a marginal density of properties $p(\boldsymbol{\kappa}|\boldsymbol{\varphi}) = \mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{\varphi})}[p(\boldsymbol{\kappa}|\boldsymbol{x})]$ that is as close as possible to this target (Fig. 2c). While there are several distance measures in the space of densities, we employ here the Kullback–Leibler divergence $KL(p_{target}(\boldsymbol{\kappa})||p(\boldsymbol{\kappa}|\boldsymbol{\varphi}))$, the minimization of which is equivalent to (see Supplementary Notes) [Eq. 7]:

$$\begin{aligned}&(\mathrm{O2}): \qquad \boldsymbol{\varphi}^* = \arg\max_{\boldsymbol{\varphi}} U_2(\boldsymbol{\varphi})\\ &\text{where } U_2(\boldsymbol{\varphi}) = \int p_{target}(\boldsymbol{\kappa}) \log p(\boldsymbol{\kappa}|\boldsymbol{\varphi})\, d\boldsymbol{\kappa}\end{aligned} \tag{7}$$

The aforementioned objective resembles the one employed in[17], but rather than finding a density on the microstructure (or features thereof) that leads to a close match of $p_{target}(\boldsymbol{\kappa})$, we identify the processing variables $\boldsymbol{\varphi}$ that do so (i.e., we are a-priori constrained to distributions realizable for specific processing conditions $\boldsymbol{\varphi}$).

We note that both problems are considerably more challenging than deterministic counterparts, as in both cases the objectives involve expectations with respect to the high-dimensional vector(s) $\boldsymbol{x}$ (and potentially $\boldsymbol{\kappa}$), representing the microstructure (and their effective properties). Additionally, in the case of (O2), the analytically intractable density $p(\boldsymbol{\kappa}|\boldsymbol{\varphi})$ appears explicitly in the objective. While one might argue that a brute-force Monte Carlo approach with a sufficiently large number of samples would suffice to carry out the aforementioned integrations, we note that propagating the uncertainty from $\boldsymbol{x}$ to the properties $\boldsymbol{\kappa}$ would also require commensurate solutions of the expensive structure-property model which would need to be repeated for various $\boldsymbol{\varphi}$-values. To overcome challenges associated with the structure-property link, we make use of a *probabilistic* surrogate model $\mathcal{M}$ trained on data $\mathcal{D}$ with a

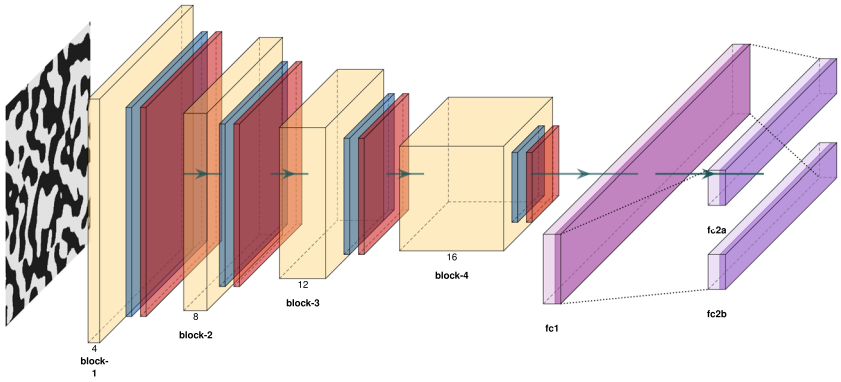


Fig. 9 Architecture of the convolutional-neural-network surrogate for property κ prediction. Features are extracted from the microstructure $\boldsymbol{x}$ using a sequence of 4 blocks (each comprised of a sequence of convolutional layer, nonlinear activation function and pooling), where in each block the size of the feature map is reduced, while the depth of the feature map increases. Fully connected feedforward layers map the extracted convolutional features to the mean $\boldsymbol{m}_{\boldsymbol{\theta}}(\boldsymbol{x})$ and the covariance $\boldsymbol{S}_{\boldsymbol{\theta}}(\boldsymbol{x})$ of the predictive Gaussian distribution $p_{\mathcal{M}}(\boldsymbol{\kappa}|\boldsymbol{x},\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\kappa}|\,\boldsymbol{m}_{\boldsymbol{\theta}}(\boldsymbol{x}),\boldsymbol{S}_{\boldsymbol{\theta}}(\boldsymbol{x}))$, where $\boldsymbol{\theta}$ denotes the neural network parameters.

predictive density $p_{\mathcal{M}}(\boldsymbol{\kappa}|\boldsymbol{x},\mathcal{D})$, which we use in place of the true $p(\boldsymbol{\kappa}|\varphi)$ in the expressions above.

We note that an alternative strategy based on circumventing the high-dimensional $\boldsymbol{x}$ and trying to approximate directly $p(\boldsymbol{\kappa}|\varphi)$[36], while tempting, will quickly become infeasible in terms of data requirements (i.e., triplets of $(\varphi,\boldsymbol{x},\boldsymbol{\kappa})$) even for modest dimensions of φ. The reformulated objectives based on $p_{\mathcal{M}}(\boldsymbol{\kappa}|\boldsymbol{x},\mathcal{D})$ are denoted with $U^{\mathcal{D}}_{1,\mathcal{M}}$ and $U^{\mathcal{D}}_{2,\mathcal{M}}$. We discuss the solution strategy of the optimization problem as well as the specifics of the probabilistic surrogate in the next sections.

Expectation-maximization and stochastic variational inference

We present the proposed algorithm for the solution of (O1) and discuss the requisite changes for (O2) afterward. The goal to identify the optimal process parameters φ^* remains challenging despite the introduction of an inexpensive, probabilistic surrogate, since the the objective functions as well as their derivatives remain intractable due to the averaging over the high-dimensional microstructures $\boldsymbol{x}$, as well as the—in the general case—intractable integration over $\boldsymbol{\kappa}$ in Eq. (5). For this reason we propose to employ the Expectation-Maximization scheme[37], which is based on the so-called Evidence Lower BOund (ELBO) $\mathcal{F}$ [Eq. 8]:

$$\begin{aligned}\log U^{\mathcal{D}}_{1,\mathcal{M}}(\varphi) &= \log\int u(\boldsymbol{\kappa})\,p_{\mathcal{M}}(\boldsymbol{\kappa}|\boldsymbol{x},\mathcal{D})\,p(\boldsymbol{x}|\varphi)\,d\boldsymbol{\kappa}\,d\boldsymbol{x}\\ &\geq \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{\kappa})}\left[\log\frac{u(\boldsymbol{\kappa})\,p_{\mathcal{M}}(\boldsymbol{\kappa}|\boldsymbol{x},\mathcal{D})\,p(\boldsymbol{x}|\varphi)}{q(\boldsymbol{x},\boldsymbol{\kappa})}\right]\\ &= \mathcal{F}(q(\boldsymbol{\kappa},\boldsymbol{x}),\varphi)\end{aligned} \tag{8}$$

where $\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{\kappa})}[.]$ denotes an expectation with respect to the auxiliary density $q(\boldsymbol{x},\boldsymbol{\kappa})$. The algorithm alternates between maximizing $\mathcal{F}$ with respect to the density $q(\boldsymbol{x},\boldsymbol{\kappa})$ while φ is fixed (E-step) and maximizing with respect to φ (M-step) while $q(\boldsymbol{x},\boldsymbol{\kappa})$ is fixed. We employ a Variational-Bayesian relaxation[38], in short VB-EM, according to which instead of the optimal q we consider a family $\mathcal{Q}_{\boldsymbol{\xi}}$ of densities parameterized by $\boldsymbol{\xi}$ and in the E-step maximize $\mathcal{F}$ with respect to $\boldsymbol{\xi}$. This, as well as the the maximization with respect to φ in the M-step, are done by using stochastic gradient ascent where the associated derivatives are substituted by noisy Monte Carlo estimates (i.e., SVI[16]). The particulars of $\boldsymbol{\xi}$ as well as of the E- and M-steps are discussed in the Supplementary Notes. We illustrate the basic, numerical steps in the inner-loop of Algorithm (1) (the algorithm starts from an initial, typically random, guess of $\boldsymbol{\xi}$ and φ). Colloquially, the VB-EM iterations can be explained as follows: In the E-step and given the current estimate for φ, one averages over microstructures that are not only a priori more probable according to $p(\boldsymbol{x}|\varphi)$ but also achieve a higher score according to $u(\boldsymbol{\kappa})\,p_{\mathcal{M}}(\boldsymbol{\kappa}|\boldsymbol{x},\mathcal{D})$. Subsequently, in the M-step step, we update the optimization variables φ on the basis of the average above (see Supplementary Notes for further details).

The second objective, $U_{2,\mathcal{M}}$ (Eq. (7)) can be dealt with in a similar fashion. The integration over $\boldsymbol{\kappa}$ with respect to the target density $p_{target}(\boldsymbol{\kappa})$ is first approximated using S Monte Carlo samples $\{\boldsymbol{\kappa}^{(s)}\}_{s=1}^{S}$ from $p_{target}(\boldsymbol{\kappa})$, and subsequently each of the terms in the sum can be lower-bounded as follows [Eq. 9]:

$$\begin{aligned}U^{\mathcal{D}}_{2,\mathcal{M}}(\varphi) &= \int p_{target}(\boldsymbol{\kappa})\log p_{\mathcal{M}}(\boldsymbol{\kappa}|\varphi,\mathcal{D})\,d\boldsymbol{\kappa}\\ &\approx \tfrac{1}{S}\textstyle\sum_{s=1}^{S}\log p_{\mathcal{M}}(\boldsymbol{\kappa}^{(s)}|\varphi,\mathcal{D})\\ &= \tfrac{1}{S}\textstyle\sum_{s=1}^{S}\log\int p_{\mathcal{M}}(\boldsymbol{\kappa}^{(s)}|\boldsymbol{x},\mathcal{D})\,p(\boldsymbol{x}|\varphi)\,d\boldsymbol{x}\\ &\geq \tfrac{1}{S}\textstyle\sum_{s=1}^{S}\mathbb{E}_{q^{(s)}(\boldsymbol{x})}\left[\log\frac{p_{\mathcal{M}}(\boldsymbol{\kappa}^{(s)}|\boldsymbol{x},\mathcal{D})\,p(\boldsymbol{x}|\varphi)}{q^{(s)}(\boldsymbol{x})}\right]\\ &= \tfrac{1}{S}\textstyle\sum_{s=1}^{S}\mathcal{F}_s\left(q^{(s)}(\boldsymbol{x}),\varphi\right)\end{aligned} \tag{9}$$

In this case, the aforementioned SVI tools will need to be applied for updating each $q^{(s)}(\boldsymbol{x}), s = 1,\ldots,S$ in the E-step, but the overall algorithm remains conceptually identical. We note that incremental and partial versions of the EM-algorithm are possible, where e.g., one or more steps of stochastic gradient ascent are performed for a subset of the $q^{(s)}$[39], leading to overall improved computational performance.

Probabilistic surrogate model

Despite the introduction of densities in the VB-EM algorithm which are tailored to the optimization problem and which enable accurate Monte Carlo estimates of the high-dimensional integrations involved, multiple evaluations of the S-P link are still required. To that end, the high-fidelity model (i.e., $\boldsymbol{\kappa}(\boldsymbol{x})$ or $p(\boldsymbol{\kappa}|\varphi)$), is substituted by a data-driven surrogate (i.e., $p_{\mathcal{M}}(\boldsymbol{\kappa}|\boldsymbol{x},\mathcal{D})$) which is trained on N pairs [Eq. 10]

$$\mathcal{D} = \left\{\boldsymbol{x}^{(n)},\boldsymbol{\kappa}^{(n)} = \boldsymbol{\kappa}\left(\boldsymbol{x}^{(n)}\right)\right\}_{n=1}^{N} \tag{10}$$

generated by the deterministic/stochastic high-fidelity model. While such supervised machine-learning problems have been studied extensively and a lot of the associated tools have found their way in materials applications[40], we note that their use in the context of the optimization problems presented requires significant adaptations.

In particular, and unlike canonical, data-centric applications relying on the abundance of data (Big Data), we operate under a smallest-possible-data regime. This is because in our setting training data arises from expensive simulations, the number of which we want to minimize. The shortage of information generally leads to predictive uncertainty (even for deterministic S-P links) which, rather than dismissing, we quantify by employing a *probabilistic surrogate* that yields a predictive density $p_{\mathcal{M}}(\boldsymbol{\kappa}|\boldsymbol{x},\mathcal{D})$ instead of mere point estimates. More importantly though, we note that the distribution of the inputs in $\mathcal{D}$, i.e., the microstructures $\boldsymbol{x}$, changes drastically with φ (Fig. 1). As we do not know a priori the optimal φ^*, we cannot generate training data from $p(\boldsymbol{\kappa}|\varphi^*)$. At the same time it is well known that data-driven surrogates produce poor extrapolative, out-of-distribution predictions[41]. It is clear therefore, that the selection of the training data, i.e., the microstructures-inputs $\boldsymbol{x}^{(n)}$ for which we pay the price of computing the output-property of interest $\boldsymbol{\kappa}^{(n)}$, should be informed by the optimization algorithm in order to produce a sufficiently accurate surrogate while keeping N as small as possible. We defer a detailed discussion of this aspect for the next section, and first present the particulars of the surrogate model employed.

The probabilistic surrogate $\mathcal{M}$ adopted has a Gaussian likelihood, i.e., $p_{\mathcal{M}}(\boldsymbol{\kappa}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{\kappa}|\boldsymbol{m}_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{S}_{\boldsymbol{\theta}}(\boldsymbol{x}))$, where the mean $\boldsymbol{m}_{\boldsymbol{\theta}}(\boldsymbol{x})$ and covariance $\boldsymbol{S}_{\boldsymbol{\theta}}(\boldsymbol{x})$ are modeled with a convolutional neural network (CNN) (see Fig. 9 and Supplementary Notes for more details), with $\boldsymbol{\theta}$ denoting the associated neural network parameters. CNNs have been used previously for property prediction in binary media in e.g.,[42,43]. Point estimates $\boldsymbol{\theta}_{\mathcal{D}}$ of the parameters are obtained with the help of training data $\mathcal{D}$ by maximizing the corresponding likelihood $p_{\mathcal{M}}(\mathcal{D}|\boldsymbol{\theta}) = \prod_{n=1}^{N} \mathcal{N}(\boldsymbol{\kappa}^{(n)}|\boldsymbol{m}_{\boldsymbol{\theta}}(\boldsymbol{x}^{(n)}), \boldsymbol{S}_{\boldsymbol{\theta}}(\boldsymbol{x}^{(n)}))$. On the basis of these estimates, the predictive density (i.e for a new input-microstructure $\boldsymbol{x}$) of the surrogate follows as $p_{\mathcal{M}}(\boldsymbol{\kappa}|\boldsymbol{x}, \mathcal{D}) = \mathcal{N}(\boldsymbol{\kappa}|\boldsymbol{m}_{\boldsymbol{\theta}_{\mathcal{D}}}(\boldsymbol{x}), \boldsymbol{S}_{\boldsymbol{\theta}_{\mathcal{D}}}(\boldsymbol{x}))$. We emphasize the dependence of the probabilistic surrogate on the dataset $\mathcal{D}$, for which we will discuss an adaptive acquisition strategy in the following section. While the results obtained are based on this particular architecture of the surrogate, the methodological framework proposed can accommodate any probabilistic surrogate and integrate its predictive uncertainty in the optimization procedure. Similarly, the same data-based approach could also be adopted for $p(\boldsymbol{x}|\varphi)$.

Active learning

Active learning refers to a family of methods whose goal is to improve learning accuracy and efficiency by selecting particularly salient training data[44]. This is especially relevant in our application, in which the acquisition of data is de facto the most computationally expensive component. The basis of all such methods is to progressively enrich the training dataset by scoring candidate inputs (i.e., microstructures $\boldsymbol{x}$ in our case) based on their *expected informativeness*[45]. The latter can be quantified with a so-called acquisition function $a(\boldsymbol{x})$, for which many different forms have been proposed (depending on the specific setting). We note though that in most cases in the literature, acquisition functions associated with the predictive accuracy of the supervised learning model have been employed, which in our formulation translates to the accuracy of our surrogate in predicting the properties $\boldsymbol{\kappa}$ for an input-microstructure. Alternate acquisition functions have been proposed in the context of Bayesian Optimization problems which as explained in the introduction exhibit significant differences with ours[11]. While it is true that a *perfect* surrogate (i.e., if $p(\boldsymbol{\kappa}|\boldsymbol{x}) = p_{\mathcal{M}}(\boldsymbol{\kappa}|\boldsymbol{x}, \mathcal{D})\,\forall \boldsymbol{x}$) would yield the exact optimum, this is not a necessary condition. An *approximate* surrogate is sufficient, as long as its aggregate predictions can correctly guide the search in the φ-space in order to discover the optimal value of φ for (O1) or (O2). This also implies that an accurate surrogate for $\varphi-$values (and corresponding microstructures $\boldsymbol{x}$) far away from the optimum is not necessary. The difficulty of course is that we do not know a priori what is the optimum φ^* and a surrogate trained on microstructures drawn from $p(\boldsymbol{x}|\varphi^{(0)})$ (with $\varphi^{(0)}$ being the initial guess in the optimization—see Algorithm (1)) will generally perform poorly at other φ's.

The acquisition function that we propose incorporates the optimization objectives. In particular, for the (O1) problem (Eq. (5)) it is given by:

$$a(\boldsymbol{x}) = \mathrm{Var}_{p_{\mathcal{M}}(\boldsymbol{\kappa}|\boldsymbol{x},\mathcal{D})}[u(\boldsymbol{\kappa})] \tag{11}$$

We note that a scores each microstructure $\boldsymbol{x}$ in terms of the predictive uncertainty in the utility u (the expected value of which we seek to maximize) due to the predictive density of the surrogate. In the case discussed earlier where $u(\boldsymbol{\kappa}) = \mathbb{I}_{\mathcal{K}}(\boldsymbol{\kappa})$ (and $U_1(\varphi) = Pr(\boldsymbol{\kappa} \in \mathcal{K}|\varphi)$), the acquisition function reduces to the variance of the event $\boldsymbol{\kappa} \in \mathcal{K}$. This suggests that the acquisition function yields the largest scores for microstructures for which the surrogate is most uncertain whether their corresponding properties fall within the target domain $\mathcal{K}$.

We propose a general procedure according to which the VB-EM-based optimization is embedded in an outer loop indexed by the data augmentation steps $l = 1, \ldots L$. Hence $\mathcal{D}^{(l)}$ denotes the training dataset at step l, $p_{\mathcal{M}}(\boldsymbol{\kappa}|\boldsymbol{x}, \mathcal{D}^{(l)})$ the corresponding predictive density of the surrogate, $q^{(l)}(\boldsymbol{x})$ the marginal variational density found in the last E-step and $\varphi^{(l)}$ the optimum found in the last M-step. With this notation in hand we can then summarize the adaptive data augmentation as follows (see also Algorithm (1) in Fig. 10):

- in each outer loop iteration l we randomly generate a pool of candidate microstructures $\{\boldsymbol{x}^{(l,n)}\}_{n=1}^{N_{pool}}$ from $q^{(l)}(\boldsymbol{x})$ and select a subset of $N_{add} < N_{pool}$ microstructures which yield the highest values of the acquisition function $a(\boldsymbol{x}^{(l,n)})$.
- We solve the high-fidelity model for the aforementioned N_{add} microstructures and construct a new training dataset $D_{add}^{(l)}$ which we add to $\mathcal{D}^{(l)}$ in order to form $\mathcal{D}^{(l+1)} = \mathcal{D}^{(l)} \cup D_{add}^{(l)}$. We retrain the surrogate based on $\mathcal{D}^{(l+1)}$, i.e., we compute $p_{\mathcal{M}}(\boldsymbol{\kappa}|\boldsymbol{x}, \mathcal{D}^{(l+1)})$, and restart the VB-EM-based optimization algorithm with the updated surrogate (we note that retraining could be avoided by making use of online learning[46], which can accomodate incremental adaptions of the dataset).

For the (O2) problem we propose to select microstructures that yield the highest predictive log-score on the sample representation $\{\boldsymbol{\kappa}^{(s)}\}_{s=1}^{S}$ of the target distribution, i.e. [Eq. 12],

$$a(\boldsymbol{x}) = \frac{1}{S}\sum_{s=1}^{S} \log p_{\mathcal{M}}(\boldsymbol{\kappa}^{(s)}|\boldsymbol{x}) \qquad \boldsymbol{\kappa}^{(s)} \sim p_{target}(\boldsymbol{\kappa}) \tag{12}$$

Algorithm 1 Obtain $\varphi^* = \arg\max_{\varphi} U_{1,2}$ (Eq. (1) or Eq. (3) using a probabilistic surrogate and active learning

Data: $l = 0$, $t = 0$, $\mathcal{D}^{(0)}$, L, structure-property-model, surrogate $p\left(\boldsymbol{\kappa}\middle|\boldsymbol{x}, \mathcal{D}^{(0)}\right)$, initial $\varphi^{(0)}$, variational family $\mathcal{Q}_{\boldsymbol{\xi}}$

Result: Converged process parameter $\varphi^*_{\mathcal{D}^{(L)}}$

for $l = 1, ..., L$ **do**

  while *ELBO not converged* **do**

    /* Execute E-step */

    $$\boldsymbol{\xi}^{(t+1)} = \arg\max_{\boldsymbol{\xi}} \mathcal{F}\left(\varphi^{(t)}, q_{\boldsymbol{\xi}}(\boldsymbol{\kappa}, \boldsymbol{x})\right)$$

    /* Execute M-step */

    $$\varphi^{(t+1)} = \arg\max_{\varphi} \mathcal{F}\left(\varphi, q_{\boldsymbol{\xi}^{(t+1)}}(\boldsymbol{\kappa}, \boldsymbol{x})\right)$$

    $t \to t+1$

  end

  /* Optimal φ conditional on current data */

  $\varphi^*_{\mathcal{M},\mathcal{D}^{(l)}} \leftarrow \varphi^{(t)}$

  /* Create microstructure candidates */

  sample $\boldsymbol{x}^{(l,n)} \sim q(\boldsymbol{x})$, $n = 1, \ldots, N_{pool}$

  compute $\alpha(\boldsymbol{x}^{(l,n)})$ (Eq. (7))

  /* Select most informative subset */

  Select $N_{add} < N_{pool}$ microstructures from $\mathcal{D}^{(l)}_{pool}$ which yield the highest acquisition function values and compute the corresponding property values $\boldsymbol{\kappa}(\boldsymbol{x})$ in order to form $\mathcal{D}^{(l)}_{add}$.

  $$\mathcal{D}^{(l+1)} \leftarrow \mathcal{D}^{(l)} \cup \mathcal{D}^{(l)}_{add}$$

  /* Update probabilistic surrogate */

  $$p\left(\boldsymbol{\kappa}\middle|\boldsymbol{x}, \mathcal{D}^{(l+1)}\right) \leftarrow p\left(\boldsymbol{\kappa}\middle|\boldsymbol{x}, \mathcal{D}^{(l)}\right)$$

end

Fig. 10 Pseudo-code for proposed algorithm. The inner VB-EM iterations are wrapped within the adaptive data acquisition as an outer loop.

DATA AVAILABILITY

The accompanying data is available at https://github.com/bdevl/SMO.

CODE AVAILABILITY

The source code is available at https://github.com/bdevl/SMO.

Received: 2 August 2021; Accepted: 3 February 2022;
Published online: 21 March 2022

REFERENCES

1. National Science and Technology Council. *Materials Genome Initiative for Global Competitiveness* (Executive Office of the President, National Science and Technology Council, 2011).
2. McDowell, D. L. et al. *Integrated design of multiscale, multifunctional materials and products* (Butterworth-Heinemann, 2009).
3. Arróyave, R. & McDowell, D. L. Systems approaches to materials design: Past, present, and future. *Annu. Rev. Mater. Res.* **49**, 103–126 (2019).
4. Chernatynskiy, A., Phillpot, S. R. & LeSar, R. Uncertainty quantification in multiscale simulation of materials: a prospective. *Annu. Rev. Mater. Res.* **43**, 157–182 (2013).
5. Honarmandi, P. & Arróyave, R. Uncertainty quantification and propagation in computational materials science and simulation-assisted materials design. *Integr. Mater. Manuf. Innov.* **9**, 103–143 (2020).
6. Liu, X., Furrer, D., Kosters, J. & Holmes, J. NASA Vision 2040: A Roadmap for Integrated, Multiscale Modeling and Simulation of Materials and Systems. *Tech. Rep.* https://ntrs.nasa.gov/citations/20180002010 (2018).
7. Bock, F. E. et al. A review of the application of machine learning and data mining approaches in continuum materials mechanics. *Front. Mater.* **6**, https://www.frontiersin.org/article/10.3389/fmats.2019.00110 (2019).
8. Panchal, J. H., Kalidindi, S. R. & McDowell, D. L. Key computational modeling issues in integrated computational materials engineering. *Comput. -Aided Des.* **45**, 4–25 (2013).
9. Grigo, C. & Koutsourelakis, P.-S. Bayesian model and dimension reduction for uncertainty propagation: applications in random media. *SIAM/ASA J. Uncertain. Quantif.* **7**, 292–323 (2019).
10. Zabaras, N. & Ganapathysubramanian, B. A scalable framework for the solution of stochastic inverse problems using a sparse grid collocation approach. *J. Comput. Phys.* **227**, 4697–4735 (2008).
11. Frazier, P. I. & Wang, J. *Bayesian optimization for materials design. In Information Science for Materials Discovery and Design*, 45–75 (Springer, 2015).
12. Zhang, Y., Apley, D. W. & Chen, W. Bayesian optimization for materials design with mixed quantitative and qualitative variables. *Sci. Rep.* **10**, 4924 (2020).
13. Jung, J., Yoon, J. I., Park, H. K., Jo, H. & Kim, H. S. Microstructure design using machine learning generated low dimensional and continuous design space. *Materialia* **11**, 100690 (2020).
14. Chen, C.-T. & Gu, G. X. Machine learning for composite materials. *MRS Commun.* **9**, 556–566 (2019).
15. Torquato, S. Optimal design of heterogeneous materials. *Annu. Rev. Mater. Res.* **40**, 101–129 (2010).
16. Hoffman, M. D., Blei, D. M., Wang, C. & Paisley, J. Stochastic Variational Inference. *J. Mach. Learn. Res.* **14**, 1303–1347 (2013).
17. Tran, A. & Wildey, T. Solving stochastic inverse problems for Property–Structure linkages using data-consistent inversion and machine learning. *Jom-us.* **73**, 72–89 (2020).
18. Nosouhi Dehnavi, F., Safdari, M., Abrinia, K., Hasanabadi, A. & Baniassadi, M. A framework for optimal microstructural design of random heterogeneous materials. *Comput. Mech.* **66**, 123–139 (2020).
19. Acar, P., Srivastava, S. & Sundararaghavan, V. Stochastic design optimization of microstructures with utilization of a linear solver. *AIAA J.* **55**, 3161–3168 (2017).
20. Acar, P. & Sundararaghavan, V. Stochastic design optimization of microstructural features using linear programming for robust design. *AIAA J.* **57**, 448–455 (2019).
21. Liu, R. et al. A predictive machine learning approach for microstructure optimization and materials design. *Sci. Rep.* **5**, 1–12 (2015).
22. Paul, A. et al. Microstructure optimization with constrained design objectives using machine learning-based feedback-aware data-generation. *Nato. Sc. S. Ss. Iii. C. S.* **160**, 334–351 (2019).
23. Teubner, M. Level surfaces of Gaussian random fields and microemulsions. *Europhys. Lett.* **14**, 403–408 (1991).
24. Roberts, A. P. & Teubner, M. Transport properties of heterogeneous materials derived from Gaussian random fields: Bounds and simulation. *Phys. Rev. E* **51**, 4141–4154 (1995).
25. Koutsourelakis, P. Probabilistic characterization and simulation of multi-phase random media. *Probabilist. Eng. Mech.* **21**, 227–234 (2006).
26. Bostanabad, R., Bui, A. T., Xie, W., Apley, D. W. & Chen, W. Stochastic microstructure characterization and reconstruction via supervised learning. *Acta Mater.* **103**, 89–102 (2016).
27. Cang, R. et al. Microstructure representation and reconstruction of heterogeneous materials via deep belief network for computational material design. *J. Mech. Design* **139**, https://asmedigitalcollection.asme.org/mechanicaldesign/articleabstract/139/7/071404/383783/Microstructure-Representation-and-Reconstruction (2017).
28. Miehe, C. & Koch, A. Computational micro-to-macro transitions of discretized microstructures undergoing small strains. *Arch. Appl. Mech.* **72**, 300–317 (2002).
29. Hill, R. On constitutive macro-variables for heterogeneous solids at finite strain. *Proc. R. Soc. A: Math. Phys. Eng. Sci.* **326**, 131–147 (1972).
30. Saheli, G., Garmestani, H. & Adams, B. L. Microstructure design of a two phase composite using two-point correlation functions. *J. Comput. -Aided Mater. Des.* **11**, 103–115 (2004).
31. Fullwood, D. T., Niezgoda, S. R., Adams, B. L. & Kalidindi, S. R. Microstructure sensitive design for performance optimization. *Prog. Mater. Sci.* **55**, 477–562 (2010).
32. Sternfels, R. & Koutsourelakis, P.-S. Stochastic design and control in random heterogeneous materials. *Int. J. Multiscale Com.* **9**, 425–443 (2011).
33. Popova, E. et al. Process-structure linkages using a data science approach: application to simulated additive manufacturing data. *Integr. Mater. Manuf. Innov.* **6**, 54–68 (2017).
34. Lee, X. Y. et al. Fast inverse design of microstructures via generative invariance networks. *Nat. Comput. Sci.* **1**, 229–238 (2021).
35. Ikebata, H., Hongo, K., Isomura, T., Maezono, R. & Yoshida, R. Bayesian molecular design with a chemical language model. *J. Comput. Aided Mol. Des.* **31**, 379–391 (2017).
36. Tran, A. & Wildey, T. Solving stochastic inverse problems for Property–Structure linkages using data-consistent inversion and machine learning. *Jom-us.* **73**, 72–89 (2020).
37. Dempster, A. P., Laird, N. M. & Rubin, D. B. Maximum likelihood from incomplete data via the EM algorithm (with discussion). *J. R. Stat. Soc. Ser. B* **39**, 1–38 (1977).
38. Beal, M. J. & Ghahramani, Z. Variational Bayesian learning of directed graphical models with hidden variables. *Bayesian Anal.* **1**, 793–832 (2006).
39. Neal, R. M. & Hinton, G. E. A view of the em algorithm that justifies incremental, sparse, and other variants. In: *Learning in Graphical Models*, MIT Press, 355–368 (1998).
40. Kalidindi, S. R. A Bayesian framework for materials knowledge systems. *MRS Commun.* **9**, 518–531 (2019).
41. Marcus, G. & Davis, E. *Rebooting AI: Building Artificial Intelligence We Can Trust* (Vintage, 2019).
42. Yang, Z. et al. Deep learning approaches for mining structure-property linkages in high contrast composites from simulation datasets. *Nato. Sc. S. Ss. Iii. C. S.* **151**, 278–287 (2018).
43. Cecen, A., Dai, H., Yabansu, Y. C., Kalidindi, S. R. & Song, L. Material structure-property linkages using three-dimensional convolutional neural networks. *Acta Mater.* **146**, 76–84 (2018).
44. Tong, S. *Active Learning: Theory and Applications*. Dissertation, Stanford University. https://scholar.google.de/scholar?hl=de&as_sdt=0%2C5&q=Active+learning%3A+Theory+and+applications&btnG (2001).
45. MacKay, D. J. C. Information-based objective functions for active data selection. *Neural Comput.* **4**, 590–604 (1992).
46. Sahoo, D., Pham, Q., Lu, J. & Hoi, S. C. H. Online deep learning: learning deep neural networks on the fly. In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence Organization*, 2660–2666. https://www.ijcai.org/proceedings/2018/369 (2018).

ACKNOWLEDGEMENTS

Funded under the Excellence Strategy of the Federal Government and the Länder in the context of the ARTEMIS Innovation Network.

AUTHOR CONTRIBUTIONS

M.R.: conceptualization, physics and machine-learning modeling and computations, algorithmic and code development, writing of the paper. P-S.K: conceptualization, writing of the paper.

FUNDING

Open Access funding enabled and organized by Projekt DEAL.

COMPETING INTERESTS

The authors declare no competing interests.

ADDITIONAL INFORMATION

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41524-022-00718-6.

Correspondence and requests for materials should be addressed to Phaedon-Stelios Koutsourelakis.

Supplementary Material: Self-supervised optimization of random material microstructures in the small-data regime

Maximilian Rixner[1] and Phaedon-Stelios Koutsourelakis[*,1,2]

[1] *Technical University of Munich, Department of Engineering Physics and Computation, Professorship of Data-driven Materials Modeling*
[2] *Munich Data Science Institute (MDSI - Core Member)*

[*] *Corresponding author, p.s.koutsourelakis@tum.de*

Supplementary Note

In the following, we provide details specific to the algorithmic implementation and the numerical simulations.

1.1 Process-Structure linkage

As mentioned in the main text, the discretized, two-phase random microstructures employed in the numerical illustrations are represented by a random vector $\boldsymbol{x}$ which arises by thresholding a two-dimensional, zero-mean, unit-variance Gaussian field, in its discretized form denoted by the vector $\boldsymbol{x}_g$. The cutoff threshold x_0 is specified by the desired volume fraction and the parameters φ are associated with the spectral density function (SDF) $G(\boldsymbol{w})$ of the underlying Gaussian field. The SDF $G(\boldsymbol{w})$ arises as the Fourier dual of the autocovariance, where $\boldsymbol{w} = [w_1, w_2]^T \in \mathbb{R}^2$ denotes the wavenumbers. We express the SDF as:

$$G(\boldsymbol{w}) = \sum_{i=1}^{Q} \gamma_i h_i(\boldsymbol{w}; \boldsymbol{\mu}_i, \sigma_i) \tag{1}$$

where the functions h_i are Radial Basis Functions (RBFs) which depend on the parameters $\boldsymbol{\mu}_i, \sigma_i$ and have the functional form:

$$h(\boldsymbol{w}; \boldsymbol{\mu}, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}||\boldsymbol{w}-\boldsymbol{\mu}||^2} \tag{2}$$

This form is adopted because it automatically ensures the positivity of the resulting SDF. Eq. (1) is also known as a *spectral mixture kernel*, which defines a universal approximator for sufficiently large Q [1]. In our simulations, the parameters $\{\boldsymbol{\mu}_i\}$, i.e. the centers of RBFs, were fixed to a uniform grid in $[0, w_{max}]^2$, with $w_{max} = 65.0$ and $\sigma_i = 12.0,\ \forall i$. Finally the weights γ_i are related to the optimization variables φ through a softmax transformation:

$$\gamma_i = \frac{e^{\varphi_i}}{\sum_{j=1}^{Q} e^{\varphi_j}} \tag{3}$$

This is employed so that the resulting SDF integrates to 1 which is the variance of the corresponding Gaussian field. We made use of a spectral representation of the underlying Gaussian field (and therefore of $\boldsymbol{x}_g$) on the basis of its φ-controlled SDF and according to the formulations detailed in [2, 3, 4]. The thresholded Gaussian vector $\boldsymbol{x}_g$ gives rise to the binary microstructure $\boldsymbol{x}$ as described above and we denote summarily the corresponding transformation as:

$$\boldsymbol{x} = \boldsymbol{F}_\varphi(\boldsymbol{\Psi}) \tag{4}$$

where $\boldsymbol{\Psi}$ denotes a vector of so-called random phase angles [2]. It consists of independent random variables uniformly distributed in $[0, 2\pi]$, and its dimension depends on the discretization of the spectral domain. A direct implication of Eq. (4) is that the process-structure density $p(\boldsymbol{x}|\varphi)$ can now be expressed as:

$$p(\boldsymbol{x}|\varphi) = \int \delta(\boldsymbol{x} - \boldsymbol{F}_\varphi(\boldsymbol{\Psi}))\ p(\boldsymbol{\Psi})\ d\boldsymbol{\Psi} \tag{5}$$

where $p(\boldsymbol{\Psi})$ is the product of uniform densities $\mathcal{U}[0, 2\pi]$. As a result, expectations of arbitrary functions, say $f(\boldsymbol{x})$, with respect to $p(\boldsymbol{x}|\varphi)$ can now be written as:

$$\mathbb{E}_{p(\boldsymbol{x}|\varphi)}[f(\boldsymbol{x})] = \int f(\boldsymbol{x})\, \delta(\boldsymbol{x} - \boldsymbol{F}_\varphi(\boldsymbol{\Psi}))\, p(\boldsymbol{\Psi})\ \mathrm{d}\boldsymbol{x}\ \mathrm{d}\boldsymbol{\Psi} \tag{6}$$

$$= \int f(\boldsymbol{F}_\varphi(\boldsymbol{\Psi}))\, p(\boldsymbol{\Psi})\, \mathrm{d}\boldsymbol{\Psi} \tag{7}$$

1.2 VB-EM-Algorithm

By making use of Eq. (7) above, we can write the ELBO for the log-expected utility (type (O1) problems) as

$$\begin{aligned}
\log U^{\mathcal{D}}_{1,\mathcal{M}}(\varphi) &= \log \mathbb{E}_{p(\boldsymbol{x}|\varphi)} \left[\int u(\boldsymbol{\kappa})\, p_{\mathcal{M}}(\boldsymbol{\kappa}|\boldsymbol{x}, \mathcal{D})\, \mathrm{d}\boldsymbol{\kappa} \right] \\
&= \log \int u(\boldsymbol{\kappa})\, p_{\mathcal{M}}(\boldsymbol{\kappa}|\boldsymbol{F}_\varphi(\boldsymbol{\Psi}), \mathcal{D})\, p(\boldsymbol{\Psi})\, \mathrm{d}\boldsymbol{\kappa}\, \mathrm{d}\boldsymbol{\Psi} \\
&\geq \mathbb{E}_{q_{\boldsymbol{\xi}}(\boldsymbol{\kappa}, \boldsymbol{\Psi})} \left[\log \frac{u(\boldsymbol{\kappa})\, p_{\mathcal{M}}(\boldsymbol{\kappa}|\boldsymbol{F}_\varphi(\boldsymbol{\Psi}), \mathcal{D})\, p(\boldsymbol{\Psi})}{q_{\boldsymbol{\xi}}(\boldsymbol{\kappa}, \boldsymbol{\Psi})} \right] \\
&= \mathcal{F}(\varphi, q_{\boldsymbol{\xi}}(\boldsymbol{\kappa}, \boldsymbol{\Psi}))
\end{aligned} \tag{8}$$

where expectations with respect to $\boldsymbol{x}$ have been substituted by integrations with respect to the (primal) random variables $\boldsymbol{\Psi}$ arising from the spectral representation. Similarly the variational density is expressed with respect to $\boldsymbol{\Psi}$, i.e. $q_{\boldsymbol{\xi}}(\boldsymbol{\kappa}, \boldsymbol{\Psi})$ (as opposed to $q_{\boldsymbol{\xi}}(\boldsymbol{\kappa}, \boldsymbol{x})$). The ELBO of the (O2)-type problems can similarly be written as

$$\begin{aligned}
\log U^{\mathcal{D}}_{2,\mathcal{M}}(\varphi) &= \int p_{target}(\boldsymbol{\kappa}) \log p_{\mathcal{M}}(\boldsymbol{\kappa}|\varphi, \mathcal{D})\, \mathrm{d}\boldsymbol{\kappa} \\
&\approx \frac{1}{S} \sum_{s=1}^{S} \log p_{\mathcal{M}}\left(\boldsymbol{\kappa}^{(s)} \middle| \varphi, \mathcal{D}\right) \qquad \boldsymbol{\kappa}^{(s)} \overset{\text{i.i.d.}}{\sim} p_{target}(\boldsymbol{\kappa}) \\
&= \frac{1}{S} \sum_{s=1}^{S} \log \int p_{\mathcal{M}}\left(\boldsymbol{\kappa}^{(s)} \middle| \boldsymbol{F}_\varphi(\boldsymbol{\Psi}), \mathcal{D}\right) p(\boldsymbol{\Psi})\, \mathrm{d}\boldsymbol{\Psi} \\
&\geq \frac{1}{S} \sum_{s=1}^{S} \mathbb{E}_{q^{(s)}_{\boldsymbol{\xi}}(\boldsymbol{\Psi})} \left[\log \frac{p_{\mathcal{M}}\left(\boldsymbol{\kappa}^{(s)} \middle| \boldsymbol{F}_\varphi(\boldsymbol{\Psi}), \mathcal{D}\right) p(\boldsymbol{\Psi})}{q^{(s)}_{\boldsymbol{\xi}}(\boldsymbol{\Psi})} \right] \\
&= \sum_{s=1}^{S} \mathcal{F}_s\left(q^{(s)}_{\boldsymbol{\xi}}(\boldsymbol{\Psi}), \varphi\right)
\end{aligned} \tag{9}$$

The expression for $\log U^{\mathcal{D}}_{2,\mathcal{M}}(\varphi)$ follows from the observation that minimization of the Kullback-Leibler divergence is equivalent to maximization of $\int p_{target}(\boldsymbol{\kappa}) \log p_{\mathcal{M}}(\boldsymbol{\kappa}|\varphi, \mathcal{D})\, \mathrm{d}\boldsymbol{\kappa}$ due to the following relations:

$$\begin{aligned}
&KL(p_{target}(\boldsymbol{\kappa})||p_{\mathcal{M}}(\boldsymbol{\kappa}|\varphi, \mathcal{D})) \\
&= \mathbb{E}_{p_{target}(\boldsymbol{\kappa})} \left[\log \frac{p_{target}(\boldsymbol{\kappa})}{p_{\mathcal{M}}(\boldsymbol{\kappa}|\varphi, \mathcal{D})} \right] \\
&= -\mathbb{E}_{p_{target}(\boldsymbol{\kappa})} [\log p_{\mathcal{M}}(\boldsymbol{\kappa}|\varphi, \mathcal{D})] - \underbrace{\mathbb{H}[p_{target}(\boldsymbol{\kappa})]}_{\text{independent of } \varphi}
\end{aligned} \tag{10}$$

and consequently

$$\begin{aligned}
\varphi^* &= \arg\min_{\varphi} KL(p_{target}(\boldsymbol{\kappa})||p_{\mathcal{M}}(\boldsymbol{\kappa}|\varphi, \mathcal{D})) \\
&= \arg\max_{\varphi} \mathbb{E}_{p_{target}(\boldsymbol{\kappa})} [\log p_{\mathcal{M}}(\boldsymbol{\kappa}|\varphi, \mathcal{D})] \\
&= \arg\max U^{\mathcal{D}}_{2,\mathcal{M}}(\varphi)
\end{aligned} \tag{11}$$

Since the maximization of the ELBO w.r.t. $\boldsymbol{\xi}$ and φ is not possible in closed form, noisy estimates of the gradients $\hat{\boldsymbol{g}}_\varphi \approx \nabla_\varphi \mathcal{F}(\varphi, q_{\boldsymbol{\xi}}(\boldsymbol{\kappa}, \boldsymbol{\Psi}))$ and $\hat{\boldsymbol{g}}_{\boldsymbol{\xi}} \approx \nabla_{\boldsymbol{\xi}} \mathcal{F}(\varphi, q_{\boldsymbol{\xi}}(\boldsymbol{\kappa}, \boldsymbol{\Psi}))$ are obtained using Monte Carlo and the reparametrization trick ([5] - see ensuing

discussion). For our numerical illustrations, the optimization with respect to φ and $\boldsymbol{\xi}$ is carried out with stochastic gradient ascent and the Adam optimizer [6] in PyTorch [7].

Representation

Instead of the bounded phase angles $\boldsymbol{\Psi}$, we employ the unbounded and normally distributed variables $\boldsymbol{\Psi}_t$ (i.e. $\boldsymbol{\Psi}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$) which are related through the error function $\mathrm{erf}(\cdot)$ as follows:

$$\Psi_i = 0.5 \cdot \left(1 + \mathrm{erf}\left(\frac{\Psi_{t,i}}{\sqrt{2\pi}}\right)\right) \cdot 2\pi \tag{12}$$

As a result, all expectations with respect to $p(\boldsymbol{\Psi})$ are substituted by expectations with respect to $p(\boldsymbol{\Psi}_t) = \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.

In addition, instead of the Heaviside function defining the binary vector $\boldsymbol{x}$ from the underlying Gaussian $\boldsymbol{x}_g$ as $x_i = H(x_{g,i} - x_0)$, we employ the differentiable transformation

$$x_i = \frac{\tanh\left(\epsilon\left(x_{g,i} - x_0\right)\right) + 1}{2} \tag{13}$$

We note that as $\epsilon \to \infty$ we recover the Heaviside function and therefore a hard truncation. While the resulting x_i's are only approximately binary for finite ε (we used $\varepsilon = 25$), these were used in all computations involved in the surrogate and the optimization. An advantage is that this enables the use of the reparametrization trick [5] to estimate the ELBO and its gradients as explained in the previous section.

Low-rank Variational Approximation

For $q_{\boldsymbol{\xi}}(\boldsymbol{\kappa}, \boldsymbol{\Psi}_t) \in \mathcal{Q}$ we adopt the choice of a low-rank multivariate Gaussian distribution (with $\boldsymbol{z} = [\boldsymbol{\kappa}, \boldsymbol{\Psi}_t]^T \in \mathbb{R}^{d_z}$), i.e.

$$q_{\boldsymbol{\xi}}(\boldsymbol{z}) = \mathcal{N}\left(\boldsymbol{z} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma} = \mathrm{diag}(\boldsymbol{d}) + \boldsymbol{L}\boldsymbol{L}^T\right) \tag{14}$$

with $\boldsymbol{L} \in \mathbb{R}^{d_z \times M}$ and $M << d_z$. The variational parameters are given by $\boldsymbol{\xi} = \{\boldsymbol{\mu}, \boldsymbol{d}, \boldsymbol{L}\}$ with $\dim(\boldsymbol{\xi}) = \mathcal{O}(d_z \cdot M)$. This particular choice enables to capture enough of the correlation structure to drive the EM updates, while remaining scalable with regards to the (generally large) dimension d_z of the problem (we used $M = 50$). We note that a fully Bayesian treatment of the surrogate could be accomplished by including the neural network parameters $\boldsymbol{\theta}$ in the variational inference framework. While we could also drive the EM-algorithm via, e.g., Markov Chain Monte Carlo or Sequential Monte Carlo, the choice of variational inference is computationally faster and additionally enables monitoring of convergence through the ELBO $\mathcal{F}$.

Tempering

When specifying the material design objective, it is numerically advantageous to pursue a tempering schedule, in particular if the desired material behaviour deviates strongly from the initially observed dataset $\mathcal{D}$, or the properties $\boldsymbol{\kappa}$ associated with the initial guess $\varphi^{(0)}$. In the following we discuss an adaptive tempering strategy which - for the sake of illustration - we explain in the context of a utility function $u(\boldsymbol{\kappa}) = \mathbb{I}_{\mathcal{K}}(\boldsymbol{\kappa})$. Instead of trying to obtain $\varphi^* = \arg\max p(\boldsymbol{\kappa} \in \mathcal{K}|\varphi)$ directly, we instead introduce a sequence of target domains $\mathcal{K}^{(r)}, r = 1..., R$, such that $\mathcal{K}^{(R)} = \mathcal{K}$. To this end we may define $\mathcal{K}^{(0)}$ in such a way, that (according to the model belief) a non-negligible number of samples $\boldsymbol{\kappa} \sim p_{\mathcal{M}}\left(\boldsymbol{\kappa}\middle|\varphi^{(0)}, \mathcal{D}\right)$ fall into the domain $\mathcal{K}^{(0)}$. In order to assess how strongly the tempered target domain $\mathcal{K}^{(r)}$ can be shifted towards the desired $\mathcal{K}$ in each each step r, we can make use of the *effective sample size* (ESS) [8]. For an ensemble of N_w phase angles $\{\boldsymbol{\Psi}^{(n)}\}_{n=1}^{N_w}$ generated from $q(\boldsymbol{\Psi})$, we introduce the corresponding weights as the model-based belief that the material properties $\boldsymbol{\kappa}$ reside in the tempered target domain $\mathcal{K}^{(r)}$

$$w_n^r = \int \mathbb{I}_{\mathcal{K}^{(r)}}(\boldsymbol{\kappa})\, p_{\mathcal{M}}\left(\boldsymbol{\kappa}\middle|\boldsymbol{F}_\varphi\left(\boldsymbol{\Psi}_t^{(n)}\right), \mathcal{D}\right) \mathrm{d}\boldsymbol{\kappa} \tag{15}$$

Denoting the normalized weights as $\tilde{w}_n^r = w_n^r / \left(\sum_{n=1}^{N_w} w_n^r\right)$, the ESS is defined as

$$\mathrm{ESS} = \left(\sum_{n=1}^{N_w} {\tilde{w}_n^r}^2\right)^{-1} = \frac{\left(\sum_{n=1}^{N_w} w_n^r\right)^2}{\sum_{n=1}^{N_w} {w_n^r}^2} \tag{16}$$

where $\mathrm{ESS} \in [0, 1]$ represents the deterioration of sample quality induced by shifting the domain $\mathcal{K}^{(r)}$. Let $q(\boldsymbol{\Psi}_t)$ be an approximation to the posterior over the phase angles conditional on the optimality criteria (i.e. $\mathcal{K}^{(r)}$) and the current value of φ. One may then adaptively chose to shift the target domain $\mathcal{K}^{(r)} \to \mathcal{K}^{(r+1)}$ in such a manner, that the ESS of the samples generated from $q(\boldsymbol{\Psi}_t)$ does not deteriorate beyond a certain threshold value (e.g. using a bisection approach). When defining the material design objective by means of a target distribution $p_{target}(\boldsymbol{\kappa})$, similarly one may introduce tempering by gradually shifting the sample representation giving rise to the evidence lower bound.

1.3 Structure-Property linkage

Probabilistic Surrogate

The probabilistic surrogate employed is based on a parametric convolutional neural network (see illustration in main text), where a split in the final dense layers gives rise to (separately) the mean vector $\boldsymbol{m}_{\boldsymbol{\theta}}(\boldsymbol{x})$, as well as the covariance matrix $\boldsymbol{S}_{\boldsymbol{\theta}}(\boldsymbol{x})$ (assumed to be diagonal). The specific choices made regarding the neural network architecture are based on prior published work (e.g. [9, 10]). Each block displayed in the illustration corresponds to 2d convolutions with a subsequent non-linear activation function (Leaky ReLU) ,followed by average pooling. The convolutional layers employ a (3×3) kernel, which in combination with appropriate padding leaves the size of the feature map unchanged [1]. The subsequent average pooling always employs a (2×2) kernel (and identical stride), such that the size of the feature maps is reduced by half in each block. For the numerical results presented, after a sequence of 4 such blocks (with an increasing depth of $4, 8, 12$ and 16 channels in the feature maps), the resulting feature representation extracted from the microstructure is flattened and enters first a shared hidden layer (of width 30), subsequently splitting up into two more layers that map to the mean $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ and the diagonal covariance matrix $\boldsymbol{S}_{\boldsymbol{\theta}}$ (the positivity of the latter is ensured via an exponential transformation). The two phases were encoded as a $(+1)$ and (-1) for the CNN, as this is numerically more expedient compared to an $\{1, 0\}$ representation of the phases. For all numerical experiments presented, the neural network was trained with a batch size of $N_{bs} = 128$. To add regularization, a weight decay of 10^{-5} was used, and additionally a dropout layer (with $p = 0.05$) was introduced before the first dense layer. The neural network was trained on the log-likelihood of the data making again use of the Adam optimizer for the stochastic updates of the parameters $\boldsymbol{\theta}$.

Physical model for the computation of properties $\boldsymbol{\kappa}$

In the following we provide a more detailed description of the physical models in the structure-property linkage abstractly represented as $\boldsymbol{\kappa}(\boldsymbol{x})$, i.e., the link between the microstructures and their physical properties we want to control in this study. Note that the specific choice of $\boldsymbol{\kappa}(\boldsymbol{x})$ does not have any direct bearing on the optimization, as the structure-property linkage only enters into the generation of the training data $\mathcal{D}$ for the surrogate. For our numerical illustrations we make use of both the effective *thermal* as well as *mechanical* properties of the microstructures $\boldsymbol{x} \sim p(\boldsymbol{x}|\varphi)$. We present details regarding the numerical computation of the effective mechanical properties, with the thermal properties following by analogy. In order to quantify the macroscopic response of a microstructure, we consider a linear, isotropic elasticity problem on the microscopic scale for a representative volume element (RVE). At this scale the behaviour of the microstructure is characterized by the balance equation (BE) and constitute equation (CE)

$$\text{(BE):} \quad \mathrm{div}(\boldsymbol{\sigma}) = \mathbf{0} \qquad \forall \boldsymbol{s} \in \Omega_{\mathrm{RVE}} \tag{17}$$

$$\text{(CE):} \quad \boldsymbol{\sigma} = \mathbb{C}(\boldsymbol{s}) : \boldsymbol{\epsilon} \qquad \forall \boldsymbol{s} \in \Omega_{\mathrm{RVE}} \tag{18}$$

[1] The discretized microstructures regarded as a vectors $\boldsymbol{x} \in \{0, 1\}^{4096}$ are of course reshaped into their original (64×64) un-flattened tensor representation for the CNN.

where $\boldsymbol{\sigma}, \boldsymbol{\epsilon}$ denote microscopic stress and strain, while $\mathbb{C}(\boldsymbol{s})$ constitutes the heterogeneous elasticity tensor $\mathbb{C}(\boldsymbol{s})$. For a binary microstructures where the two phases occupy (random) subdomains $\mathcal{V}_0$ and $\mathcal{V}_1$ (with $\mathcal{V}_0 \cap \mathcal{V}_1 = \emptyset$ and $\mathcal{V}_0 \cup \mathcal{V}_1 = \Omega_{\text{RVE}}$) the elasticity tensor follows as:

$$\mathbb{C}(\boldsymbol{s}) = \begin{cases} \mathbb{C}_1, & \text{if } \boldsymbol{s} \in \mathcal{V}_1 \\ \mathbb{C}_0, & \text{if } \boldsymbol{s} \in \mathcal{V}_0 \end{cases} \tag{19}$$

In our case, the elasticity tensors $\mathbb{C}_0$ and $\mathbb{C}_1$ are fully defined by the Young's moduli E_0 and E_1 of the two phases (a common Poisson's ratio of $\nu = 0.3$ was used). We define the *macroscopic* stress $\boldsymbol{\Sigma} = \langle \boldsymbol{\sigma} \rangle$ as well as macroscopic strain $\boldsymbol{E} = \langle \boldsymbol{\epsilon} \rangle$, where $\langle \cdot \rangle$ denotes a *spatial* average of microscopic quantities over Ω_{RVE}. We then characterize the macroscopic, effective behaviour of the microstructure via [11]

$$\boldsymbol{\Sigma} = \mathbb{C}^{\text{eff}} : \boldsymbol{E} \tag{20}$$

under the constraint that (20) satisfies the averaging theorem by Hill [11, 12]

$$\boldsymbol{\Sigma} : \boldsymbol{E} = \frac{1}{|\Omega_{\text{RVE}}|} \int_{\partial\Omega_{\text{RVE}}} \boldsymbol{t} \cdot \boldsymbol{u} \, \mathrm{d}A \tag{21}$$

with microscopic tractions $\boldsymbol{t}$ and displacements $\boldsymbol{u}$. The homogenized properties $\boldsymbol{\kappa}$ are based on various entries of $\mathbb{C}^{\text{eff}}$ which is computed by solving an ensemble of elementary load cases, as given by the following macroscopic strain modes

$$\hat{\boldsymbol{E}}^{(1)} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad \hat{\boldsymbol{E}}^{(2)} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad \hat{\boldsymbol{E}}^{(3)} = \begin{bmatrix} 0 & 0.5 \\ 0.5 & 0 \end{bmatrix} \tag{22}$$

corresponding to either a pure tension or shear mode, such that $\mathbb{C}^{\text{eff}}$ is recovered by integrating the microscopic stress $\boldsymbol{\sigma}$ to obtain $\boldsymbol{\Sigma} = \langle \boldsymbol{\sigma} \rangle$. One possible approach of imposing these elementary strain modes is based on the introduction of periodic boundary condition (as opposed to defining load cases based on displacement or tractions), which augments Eq. (17) and (18) by

$$\boldsymbol{\epsilon} = \hat{\boldsymbol{E}}^{(c)} + \nabla_{\boldsymbol{s}} \boldsymbol{v} \qquad \text{in } \Omega_{\text{RVE}} \tag{23}$$

$$\boldsymbol{v} \qquad \text{is } \Omega_{\text{RVE}}\text{-periodic} \tag{24}$$

$$\boldsymbol{t} = \boldsymbol{\sigma} \cdot \boldsymbol{n} \qquad \text{is } \Omega_{\text{RVE}}\text{-antiperiodic} \tag{25}$$

Here $\boldsymbol{v}$ denotes a periodic fluctuation (i.e., $\boldsymbol{u} = \boldsymbol{E}\boldsymbol{s} + \boldsymbol{v}$), and $\boldsymbol{t}$ are antiperiodic tractions on the boundary of the domain Ω_{RVE}. We solve for the periodic fluctuations $\boldsymbol{v}$ for all three elementary load cases $c = \{1, 2, 3\}$ using the Bubnov-Galerkin approach and the standard Finite Element Method (an additional Lagrange multiplier has to be included in the variational problem to disambiguate it with regards to rigid body transformations). The effective tangent moduli $\mathbb{C}^{\text{eff}}$ of the RVE thus obtained by the solution of the differential equations (Eq. (17), (18), (23), (24) and (25)) can be shown [11] to satisfy the averaging theorem by Hill. We finally note that $\mathbb{C}^{\text{eff}}$ and the properties of interest $\boldsymbol{\kappa}$ vary depending on the underlying, random microstructure $\boldsymbol{x}$ and the need for their repeated computation represents the computational bottleneck for the optimization of the process parameters φ.

Supplementary References

[1] Wilson, A. & Adams, R. Gaussian process kernels for pattern discovery and extrapolation. In *International conference on machine learning*, 1067–1075 (PMLR, 2013).

[2] Shinozuka, M. & Deodatis, G. Simulation of Multi-Dimensional Gaussian Stochastic Fields by Spectral Representation. *Applied Mechanics Reviews* **49**, 29–53 (1996). URL https://doi.org/10.1115/1.3101883. https://asmedigitalcollection.asme.org/appliedmechanicsreviews/article-pdf/49/1/29/5437403/29_1.pdf.

[3] Hu, B. & Schiehlen, W. On the simulation of stochastic processes by spectral representation. *Probabilistic engineering mechanics* **12**, 105–113 (1997).

[4] Sternfels, R. & Koutsourelakis, P.-S. Stochastic design and control in random heterogeneous materials. *International Journal for Multiscale Computational Engineering* **9** (2011).

[5] Kingma, D. P. & Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[6] Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[7] Paszke, A. *et al.* Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* **32**, 8026–8037 (2019).

[8] Del Moral, P., Doucet, A. & Jasra, A. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society Series B-Statistical Methodology* **68**, 411–436 (2006).

[9] Yang, Z. *et al.* Deep learning approaches for mining structure-property linkages in high contrast composites from simulation datasets. *Computational Materials Science* **151**, 278–287 (2018).

[10] Cecen, A., Dai, H., Yabansu, Y. C., Kalidindi, S. R. & Song, L. Material structure-property linkages using three-dimensional convolutional neural networks. *Acta Materialia* **146**, 76–84 (2018).

[11] Miehe, C. & Koch, A. Computational micro-to-macro transitions of discretized microstructures undergoing small strains. *Archive of Applied Mechanics* **72**, 300–317 (2002).

[12] Hill, R. On constitutive macro-variables for heterogeneous solids at finite strain. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* **326**, 131–147 (1972).