Journal of Computational Physics 394 (2019) 136–152

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Adversarial uncertainty quantification in physics-informed neural networks

Yibo Yang, Paris Perdikaris *

Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania, Philadelphia, PA 19104, USA

ARTICLE INFO

Article history:
Received 9 November 2018
Received in revised form 7 March 2019
Accepted 19 May 2019
Available online 24 May 2019

Keywords:
Variational inference
Generative adversarial networks
Probabilistic deep learning
Probabilistic scientific computing
Data-driven modeling

ABSTRACT

We present a deep learning framework for quantifying and propagating uncertainty in systems governed by non-linear differential equations using physics-informed neural networks. Specifically, we employ latent variable models to construct probabilistic representations for the system states, and put forth an adversarial inference procedure for training them on data, while constraining their predictions to satisfy given physical laws expressed by partial differential equations. Such physics-informed constraints provide a regularization mechanism for effectively training deep generative models as surrogates of physical systems in which the cost of data acquisition is high, and training data-sets are typically small. This provides a flexible framework for characterizing uncertainty in the outputs of physical systems due to randomness in their inputs or noise in their observations that entirely bypasses the need for repeatedly sampling expensive experiments or numerical simulators. We demonstrate the effectiveness of our approach through a series of examples involving uncertainty propagation in non-linear conservation laws, and the discovery of constitutive laws for flow through porous media directly from noisy data.

1. Introduction

Recent advances in machine learning and data analytics have yielded transformative results across diverse scientific disciplines, including image recognition [1], natural language processing [2], cognitive science [3], and genomics [4]. In all aforementioned areas, the volume of data has increased substantially compared to even a decade ago, but analyzing big data is expensive and time-consuming. Data-driven methods, which have been enabled by the availability of sensors, data storage, and computational resources, are taking center stage across many disciplines of science. We now have highly scalable solutions for problems in object detection and recognition, machine translation, text-to-speech conversion, recommender systems, and information retrieval [2]. All of these solutions attain state-of-the-art performance when trained with large amounts of data.

However, more often than not, in laboratory experiments and large-scale simulations aiming to elucidate and predict complex phenomena, a large number of quality and error-free data is prohibitively costly to obtain. Under this setting, purely data-driven approaches for machine learning present difficulties when the data is scarce relative to the complexity of the system. The vast majority of state-of-the art machine learning techniques (e.g., deep neural nets, convolutional networks, recurrent networks, etc. [5]) are lacking robustness and fail to provide any guarantees of convergence or quantify the error/uncertainty associated with their predictions. Hence, the ability to learn in a robust and sample-efficient manner

* Corresponding author.
E-mail address: pgp@seas.upenn.edu (P. Perdikaris).

https://doi.org/10.1016/j.jcp.2019.05.027
0021-9991/

is a necessity in these data-limited domains. Even less well understood is how one can constrain such algorithms to leverage domain-specific knowledge and return predictions that satisfy certain physical principles (e.g., conservation of mass, momentum, etc.).

These shortcomings often generate skepticism and disbelief among applied mathematicians and engineers regarding the solid grounding of purely data-driven machine learning approaches. In recent work, Raissi et al. [6–10] set foot exactly at this relatively unexplored interface between applied mathematics and contemporary machine learning by revisiting the idea of penalizing the loss function of deep neural networks using differential equation constraints, as first put forth by Psichogios and Ungar [11] and Lagaris et al. [12]. This line of work has empirically demonstrated how such physics-informed constraints regularize learning in *small data* regimes, can lead to the discovery of governing equations and reduced-order models, as well as enable the prediction of complex dynamics from incomplete models and incomplete data. Despite a series of impressive results in canonical problems, Raissi et al. [6,7] have also pointed out cases in which the training phase of these algorithms faces severe difficulties for reasons that are currently poorly understood. In lack of supporting theory on convergence and a-posteriori error estimation, this naturally poses the need for scalable algorithms for uncertainty quantification.

A literature review of the current state-of-the-art in uncertainty quantification reveals a subtle dichotomy between different communities. On one hand, researchers in applied mathematics and scientific computing predominately rely on mathematical models that are rigorously derived from first physical principles. At the dawn of exascale computing, such models have enabled the accurate simulation of increasingly more complex phenomena (see for e.g., [13,14]). They have also enabled *in-silico* systematic studies in which the behavior of a system can be probed in a controlled fashion for different conditions, parameter settings, external inputs, etc. [15]. The latter aims to both elucidate the key mechanisms that govern the behavior of a system, but also characterize the robustness of the resulting predictions with respect to epistemic and aleatory uncertainty [16]. However, despite the fact that much progress has been made over the last two decades, the most popular methods for scientific computing under uncertainty, such as polynomial chaos expansions [17–19], sparse grid quadratures [20,21], multi-level/multi-fidelity Monte Carlo sampling [22,23], proper orthogonal decomposition [24,25], and Gaussian process regression models [26–28], all face severe limitations in view of the non-Gaussian likelihoods and high-dimensional posterior distributions commonly encountered in realistic applications.

On the other hand, the recent explosive growth of machine learning research has put forth new effective ways of learning and manipulating complex high-dimensional probability distributions. Inference tools like variational auto-encoders [29] and generative adversarial networks [30], formulated on top of flexible building blocks such as feed-forward/convolutional/recurrent neural networks [5] have introduced highly scalable solutions, albeit for problems where not much prior information is assumed, but instead, large amounts of data can be obtained at relatively low cost (e.g., image recognition [1], natural language processing [2]). In this work, we aim to leverage recent developments in machine learning to put forth a scalable framework for uncertainty propagation in physical systems for which the cost of data acquisition is high and training data-sets are typically small, but strong prior information exists by means of known governing laws expressed by partial differential equations. Specifically, we construct a class of probabilistic physics-informed neural networks that enables us to obtain a posterior characterization of the uncertainty associated with their predicted outputs. Moreover, we will develop a flexible variational inference framework that will allow us to train such models directly from noisy input/output data, and predict outcomes of non-linear dynamical systems that are partially observed with quantified uncertainty. This setting necessitates a departure from the classical deterministic realm of modeling and scientific computation, and, consequently, our main building blocks can no longer be crisp deterministic numbers and governing laws, but instead we must operate with *probabilistic models*.

This paper is structured as follows. In section 2.1 we provide a brief overview of physics-informed neural networks in sync with the recent developments in [6,7,9,10]. In sections 2.2 and 2.3 we provide an outline of the proposed probabilistic formulation and the proposed variational inference framework. Finally, in section 3 we will demonstrate the effectiveness of our approach through a series of examples involving uncertainty propagation in non-linear conservation laws, and the discovery of constitutive laws for flow through porous media directly from noisy data.

2. Methods

2.1. Physics-informed neural networks

The recent works of Raissi et al. [6,7,9,10,31] have demonstrated how classical conservation laws and numerical discretization schemes can be used as structured prior information that can enhance the robustness and efficiency of modern machine learning algorithms, introducing a new class of data-driven solvers, as well as a *physics-informed machine learning* approach to model discovery. To this end, the authors have considered constructing deep neural networks that return predictions which are constrained by parametrized partial differential equations (PDE) of the form

$$\boldsymbol{u}_t + \mathcal{N}_{\boldsymbol{x}}\boldsymbol{u} = 0, \tag{1}$$

where $\boldsymbol{u}(\boldsymbol{x}, t)$ is represented by a deep neural network parametrized by a set of parameters θ, i.e. $\boldsymbol{u}(\boldsymbol{x}, t) = f_\theta(\boldsymbol{x}, t)$, $\boldsymbol{x}$ is a vector of space coordinates, t is the time coordinate, and $\mathcal{N}_{\boldsymbol{x}}$ is a nonlinear differential operator. As neural networks are differentiable representations, this construction defines a so-called *physics informed neural network* that corresponds to the

PDE residual, i.e. $r_\theta(\boldsymbol{x},t) := \frac{\partial}{\partial t} f_\theta(\boldsymbol{x},t) + \mathcal{N}_{\boldsymbol{x}} f_\theta(\boldsymbol{x},t)$. This new network has the same parameters as the network representing $\boldsymbol{u}(\boldsymbol{x},t)$, albeit different activation functions due to the action of the differential operator [6,11,12]. From an implementation perspective, this network can be readily obtained by leveraging recent progress in automatic differentiation [32,33]. The resulting training procedure allows us to recover the shared network parameters θ using a few scattered observations of $\boldsymbol{u}(\boldsymbol{x},t)$, namely $\{(\boldsymbol{x}_i,t_i),\boldsymbol{u}_i\}$, $i=1,\ldots,N_u$, along with a larger number of collocation points $\{(\boldsymbol{x}_i,t_i),\boldsymbol{r}_i\}$, $i=1,\ldots,N_r$, that aim to penalize the PDE residual at a finite set of N_r collocation nodes. The data for the residual are typically zero (i.e. $\boldsymbol{r}_i=0$), or they may correspond to external forcing terms evaluated at the corresponding location $(\boldsymbol{x}_i,t_i)$, $i=1,\ldots,N_r$. This simple, yet remarkably effective regularization procedure allows us to introduce the PDE residual as a soft penalty constraint penalty in the likelihood function of the model [6,7], and the resulting optimization problem can be effectively solved using standard stochastic gradient descent without necessitating any elaborate constrained optimization techniques, simply by minimizing the composite loss function

$$\mathcal{L}(\theta) = \frac{1}{N_u}\sum_{i=1}^{N_u} \| f_\theta(\boldsymbol{x}_i,t_i) - \boldsymbol{u}_i \|^2 + \frac{1}{N_r}\sum_{i=1}^{N_r} \| r_\theta(\boldsymbol{x}_i,t_i) - \boldsymbol{r}_i \|^2, \tag{2}$$

where the required gradients $\frac{\partial \mathcal{L}}{\partial \theta}$ can be readily obtained using automatic differentiation [32]. Finally, as the resulting predictions are encouraged to inherit any physical properties imposed by the PDE constraint (e.g., conservation, invariance, symmetries, etc.), this approach showcases how one can approximately encode physical and domain-specific constraints in modern machine learning algorithms and introduce a new form of regularization for learning from *small* data-sets.

2.2. Probabilistic physics-informed neural networks

Here we put forth a probabilistic formulation for propagating uncertainty through physics-informed neural networks using latent variable models of the form

$$p(\boldsymbol{u}|\boldsymbol{x},t,\boldsymbol{z}), \quad \boldsymbol{z}\sim p(\boldsymbol{z}), \quad \text{s.t } \boldsymbol{u}_t + \mathcal{N}_{\boldsymbol{x}}\boldsymbol{u} = 0 \tag{3}$$

This setting encapsulates a wide range of deterministic and stochastic problems, where $\boldsymbol{u}(\boldsymbol{x},t)$ is a potentially multivariate random field depending on spatial and temporal variables $\boldsymbol{x}$ and t, respectively, and $\boldsymbol{z}$ is a collection of random latent variables with a prior distribution $p(\boldsymbol{z})$. The role of the latent variables $\boldsymbol{z}$ is to summarize the potentially high-dimensional factors that introduce stochasticity in the observable outputs $\boldsymbol{u}$. Here we assume that data for $\boldsymbol{u}(\boldsymbol{x},t)$ typically only corresponds to the initial/boundary conditions of the PDE, or a small set of sparse scattered measurements collected inside the domain of interest for a small number of realizations of the stochastic system. Specifically, in contrast to popular approaches to uncertainty quantification for partial differential equations, the random input factors need not be specified/observed a-priori, and their influence in introducing stochasticity in the observed outputs is distilled by the latent variables $\boldsymbol{z}$.

The ability to learn such a model from data is the cornerstone of probabilistic scientific computing and uncertainty quantification in physical systems. Knowledge of the conditional probability $p(\boldsymbol{u}|\boldsymbol{x},t,\boldsymbol{z})$ subject to domain knowledge constraints introduces a regularization mechanism that limits the space of admissible solutions to a manageable size (e.g., in fluid mechanics problems by discarding any non-realistic flow solutions that violate the conservation of mass principle), thus enables training of probabilistic deep learning models in *small data* regimes. Here, the term "small data" should be considered relative to the complexity of the underlying system. For all cases we considered in this work, the available data corresponded to $\mathcal{O}(10-1000)$ scattered measurements of the random field $\boldsymbol{u}(\boldsymbol{x},t)$. In most cases considered, the system is driven by continuous stochastic processes, and that essentially defines an infinite-dimensional problem for estimating the conditional density $p(\boldsymbol{u}|\boldsymbol{x},t)$. The notion of "small data" here is used to highlight the ability of the proposed method to tackle this challenging problem using a relatively small sets of measurements to train a physics-informed generative model that entirely bypasses the need to repeatedly sample expensive simulators in order to characterize the joint statistics of the system's response. Finally, we should also note that our probabilistic formulation enables downstream tasks such as the formulation of adaptive data acquisition policies for active learning or Bayesian optimization [34] with domain knowledge constraints.

2.3. Adversarial inference for joint distribution matching

Following the recent findings of [35] we argue that matching the joint distribution of the generated data $p_\theta(\boldsymbol{x},t,\boldsymbol{u})$ with the joint distribution of the observed data $q(\boldsymbol{x},t,\boldsymbol{u})$ by minimizing the reverse Kullback-Leibler divergence $\mathbb{KL}[p_\theta(\boldsymbol{x},t,\boldsymbol{u})||q(\boldsymbol{x},t,\boldsymbol{u})]$ is a promising approach to train the generative model presented in equation (3). This also implies that the respective marginal and conditional distributions are also encouraged to match. The use of the reverse Kullback-Leibler divergence (in contrast to the maximum likelihood setup) is motivated by examining the following decomposition

$$\mathbb{KL}[p_\theta(\boldsymbol{x},t,\boldsymbol{u})||q(\boldsymbol{x},t,\boldsymbol{u})] = -h(p_\theta(\boldsymbol{x},t,\boldsymbol{u}))) - \mathbb{E}_{p_\theta(\boldsymbol{x},t,\boldsymbol{u})}[\log(q(\boldsymbol{x},t,\boldsymbol{u}))], \tag{4}$$

where $h(p_\theta(\boldsymbol{x},t,\boldsymbol{u}))$ denotes the entropy of the generative model. The second term can be further decomposed as

$$\mathbb{E}_{p_\theta(\boldsymbol{x},t,\boldsymbol{u})}[\log(q(\boldsymbol{x},t,\boldsymbol{u}))] = \int_{\mathcal{S}_{p_\theta}\cap\mathcal{S}_q} \log(q(\boldsymbol{x},t,\boldsymbol{u}))p_\theta(\boldsymbol{x},t,\boldsymbol{u})d\boldsymbol{x}dtd\boldsymbol{u} + \int_{\mathcal{S}_{p_\theta}\cap\mathcal{S}_q^o} \log(q(\boldsymbol{x},t,\boldsymbol{u}))p_\theta(\boldsymbol{x},t,\boldsymbol{u})d\boldsymbol{x}dtd\boldsymbol{u}, \tag{5}$$

where $\mathcal{S}_{p_\theta}$ and $\mathcal{S}_q$ denote the support of the distributions $p_\theta(\boldsymbol{x},t,\boldsymbol{u})$ and $q(\boldsymbol{x},t,\boldsymbol{u})$, respectively, while $\mathcal{S}_q^o$ denotes the complement of $\mathcal{S}_q$. Notice that by minimizing the Kullback-Leibler divergence in equation (4) we introduce a mechanism that is trying to balance the effect of two competing objectives. Specifically, maximization of the entropy term $h(p_\theta(\boldsymbol{x},t,\boldsymbol{u}))$ encourages $p_\theta(\boldsymbol{x},t,\boldsymbol{u})$ to spread over its support set as wide, while the second integral term in equation (5) introduces a strong (negative) penalty when the support of $p_\theta(\boldsymbol{x},t,\boldsymbol{u})$ and $q(\boldsymbol{x},t,\boldsymbol{u})$ do not overlap. Hence, the support of $p_\theta(\boldsymbol{x},t,\boldsymbol{u})$ is encouraged to spread only up to the point that $\mathcal{S}_{p_\theta}\cap\mathcal{S}_{q^o}=\emptyset$, implying that $\mathcal{S}_{p_\theta}\subseteq\mathcal{S}_{q^o}$. When $\mathcal{S}_{p_\theta}\subset\mathcal{S}_{q^o}$ the pathological issue of "mode-collapse" (commonly encountered in the training of generative adversarial networks [30]) is manifested [36]. This issue is present if one seeks to directly minimize the reverse Kullback-Leibler objective in equation (4) as this provides no control on the relative importance of the two terms.

Notice that the decomposition in equation (5) is presented only here to motivate the constraint on the support of p_θ, and it is never practically used as part of our algorithms. The reason we spell it out is for providing intuition on how the entropy term can regularize the pathology of mode collapse. As discussed in section 2.3.3, we may rather minimize $-\lambda h(p_\theta(\boldsymbol{x},t,\boldsymbol{u})) - \mathbb{E}_{p_\theta(\boldsymbol{x},t,\boldsymbol{u})}[\log(q(\boldsymbol{x},t,\boldsymbol{u}))]$, with $\lambda \geq 1$ to allow for control of how much emphasis is placed on mitigating mode collapse. It is then clear that the entropic regularization introduced by $h(p_\theta(\boldsymbol{x},t,\boldsymbol{u}))$ provides an effective mechanism for controlling and mitigating the effect of mode collapse, and, therefore, potentially enhancing the robustness adversarial inference procedures for learning $p_\theta(\boldsymbol{x},t,\boldsymbol{u})$.

Throughout this work we have assumed that the conditional distribution $q_\phi(\boldsymbol{z}|t,\boldsymbol{x},\boldsymbol{u})$ can be modeled by a deterministic function $f_\phi(\boldsymbol{x},t,\boldsymbol{u})$, i.e. $q_\phi(\boldsymbol{z}|\boldsymbol{x},t,\boldsymbol{u}) = \delta(\boldsymbol{z} - f_\phi(\boldsymbol{x},t,\boldsymbol{u}))$, where f_ϕ is a deterministic encoder typically parametrized by a deep neural network. We found this assumption to work well in practice, although, in general, other parametrizations (e.g. Gaussian) can also be employed.

Minimization of equation (4) with respect to the generative model parameters θ presents two fundamental difficulties. First, the evaluation of both distributions $p_\theta(\boldsymbol{x},t,\boldsymbol{u})$ and $q(\boldsymbol{x},t,\boldsymbol{u})$ typically involves intractable integrals in high dimensions, and we may only have samples drawn from the two distributions, not their explicit analytical forms. Second, the differential entropy term $h(p_\theta(\boldsymbol{x},t,\boldsymbol{u}))$ is intractable as $p_\theta(\boldsymbol{x},t,\boldsymbol{u})$ is not known a-priori. In the next sections we revisit the unsupervised formulation put forth in [35] and derive a tractable inference procedure for learning $p_\theta(\boldsymbol{x},t,\boldsymbol{u})$ from scattered observation pairs of $\boldsymbol{u}(\boldsymbol{x},t)$, namely $\{(\boldsymbol{x}_i,t_i),\boldsymbol{u}_i\}$, $i=1,\ldots,N_u$.

2.3.1. Density ratio estimation by probabilistic classification

By definition, the computation of the Kullback-Leibler divergence in equation (4) involves computing an expectation over a log-density ratio, i.e.

$$\mathbb{KL}[p_\theta(\boldsymbol{x},t,\boldsymbol{u})||q(\boldsymbol{x},t,\boldsymbol{u})] := \mathbb{E}_{p_\theta(\boldsymbol{x},t,\boldsymbol{u})}\left[\log\left(\frac{p_\theta(\boldsymbol{x},t,\boldsymbol{u})}{q(\boldsymbol{x},t,\boldsymbol{u})}\right)\right].$$

In general, given samples from two distributions, we can approximate their density ratio by constructing a binary classifier that distinguishes between samples from the two distributions. To this end, we assume that N data points are drawn from $p_\theta(\boldsymbol{x},t,\boldsymbol{u})$ and are assigned a label $y=+1$. Similarly, we assume that N samples are drawn from $q(\boldsymbol{x},t,\boldsymbol{u})$ and assigned label $y=-1$. Consequently, we can write these probabilities in a conditional form, namely as

$$p_\theta(\boldsymbol{x},t,\boldsymbol{u}) = \rho(\boldsymbol{x},t,\boldsymbol{u}|y=+1), \quad q(\boldsymbol{x},t,\boldsymbol{u}) = \rho(\boldsymbol{x},t,\boldsymbol{u}|y=-1),$$

where $\rho(\boldsymbol{x},t,\boldsymbol{u}|y=+1)$ and $\rho(\boldsymbol{x},t,\boldsymbol{u}|y=-1)$ are the class probabilities predicted by a binary classifier $T(\boldsymbol{x},t,\boldsymbol{u})$. Using Bayes rule, it is then straightforward to show that the density ratio of $p_\theta(\boldsymbol{x},t,\boldsymbol{u})$ and $q(\boldsymbol{x},t,\boldsymbol{u})$ can be computed as

$$\begin{aligned}\frac{p_\theta(\boldsymbol{x},t,\boldsymbol{u})}{q(\boldsymbol{x},t,\boldsymbol{u})} &= \frac{\rho(\boldsymbol{x},t,\boldsymbol{u}|y=+1)}{\rho(\boldsymbol{x},t,\boldsymbol{u}|y=-1)} \\ &= \frac{\rho(y=+1|\boldsymbol{x},t,\boldsymbol{u})\rho(\boldsymbol{x},t,\boldsymbol{u})}{\rho(y=+1)} \Big/ \frac{\rho(y=-1|\boldsymbol{x},t,\boldsymbol{u})\rho(\boldsymbol{x},t,\boldsymbol{u})}{\rho(y=-1)} \\ &= \frac{\rho(y=+1|\boldsymbol{x},t,\boldsymbol{u})}{\rho(y=-1|\boldsymbol{x},t,\boldsymbol{u})} = \frac{\rho(y=+1|\boldsymbol{x},t,\boldsymbol{u})}{1-\rho(y=+1|\boldsymbol{x},t,\boldsymbol{u})} \\ &= \frac{T(\boldsymbol{x},t,\boldsymbol{u})}{1-T(\boldsymbol{x},t,\boldsymbol{u})},\end{aligned} \tag{6}$$

where $\rho(y=+1)=\rho(y=-1)=1/2$ and the two terms cancel if the number of the data generated by the model is balanced with the number of data originating from the true empirical distribution. This can always be done during model training as we are free to chose the number of generated data and set it equal to the size of the training data mini-batch. This is straightforward and it is a default choice in training adversarial models. Alternatively, if the number of generated versus true data is not balanced, this will simply introduce a scalar factor in equation (6). This simple procedure suggests that we can harness the power of deep neural network classifiers to obtain accurate estimates of the reverse Kullback-Leibler divergence in equation (4) directly from data and without the need to assume any specific parametrization for the generative model distribution $p_\theta(\boldsymbol{x},t,\boldsymbol{u})$.

2.3.2. Entropic regularization bound

Here we follow the derivation of Li et al. [35] to construct a computable lower bound for the entropy $h(p_\theta(\boldsymbol{x},t,\boldsymbol{u}))$. To this end, we start by considering random variables $(\boldsymbol{x},t,\boldsymbol{u},\boldsymbol{z})$ under the joint distribution

$$p_\theta(\boldsymbol{x},t,\boldsymbol{u},\boldsymbol{z})=p_\theta(\boldsymbol{u},\boldsymbol{x},t|\boldsymbol{z})p(\boldsymbol{z})=p_\theta(\boldsymbol{u}|\boldsymbol{x},t,\boldsymbol{z})p(\boldsymbol{x},t,\boldsymbol{z}),$$

where $p_\theta(\boldsymbol{u}|\boldsymbol{x},t,\boldsymbol{z})=\delta(\boldsymbol{u}-f_\theta(\boldsymbol{x},t,\boldsymbol{z}))$, and $\delta(\cdot)$ is the Dirac delta function. The mutual information between $(\boldsymbol{x},t,\boldsymbol{u})$ and $\boldsymbol{z}$ satisfies the information theoretic identity

$$I(\boldsymbol{x},t,\boldsymbol{u};\boldsymbol{z})=h(\boldsymbol{x},t,\boldsymbol{u})-h(\boldsymbol{x},t,\boldsymbol{u}|\boldsymbol{z})=h(\boldsymbol{z})-h(\boldsymbol{z}|\boldsymbol{x},t,\boldsymbol{u}),$$

where $h(\boldsymbol{x},t,\boldsymbol{u})$, $h(\boldsymbol{z})$ are the marginal entropies and $h(\boldsymbol{x},t,\boldsymbol{u}|\boldsymbol{z})$, $h(\boldsymbol{z}|\boldsymbol{x},t,\boldsymbol{u})$ are the conditional entropies [37]. Since in our setup $\boldsymbol{x}$ and t are deterministic variables independent of $\boldsymbol{z}$, and samples of $p_\theta(\boldsymbol{u}|\boldsymbol{x},t,\boldsymbol{z})$ are generated by a deterministic function $f_\theta(\boldsymbol{x},t,\boldsymbol{z})$, it follows that $h(\boldsymbol{x},t,\boldsymbol{u}|\boldsymbol{z})=0$. We therefore have

$$h(\boldsymbol{x},t,\boldsymbol{u})=h(\boldsymbol{z})-h(\boldsymbol{z}|\boldsymbol{x},t,\boldsymbol{u}), \tag{7}$$

where $h(\boldsymbol{z}):=-\int\log p(\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}$ does not depend on the generative model parameters θ.

Now consider a general variational distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x},t,\boldsymbol{u})$ parametrized by a set of parameters ϕ. Then,

$$\begin{aligned}h(\boldsymbol{z}|\boldsymbol{x},t,\boldsymbol{u})&=-\mathbb{E}_{p_\theta(\boldsymbol{x},t,\boldsymbol{u},\boldsymbol{z})}[\log(p_\theta(\boldsymbol{z}|\boldsymbol{x},t,\boldsymbol{u}))]\\&=-\mathbb{E}_{p_\theta(\boldsymbol{x},t,\boldsymbol{u},\boldsymbol{z})}[\log(q_\phi(\boldsymbol{z}|\boldsymbol{x},t,\boldsymbol{u}))]\\&\quad-\mathbb{E}_{p_\theta(\boldsymbol{x},t,\boldsymbol{u})}[\mathbb{KL}[p_\theta(\boldsymbol{z}|\boldsymbol{x},t,\boldsymbol{u})||q_\phi(\boldsymbol{z}|\boldsymbol{x},t,\boldsymbol{u})]]\\&\leq-\mathbb{E}_{p_\theta(\boldsymbol{x},t,\boldsymbol{u},\boldsymbol{z})}[\log(q_\phi(\boldsymbol{z}|\boldsymbol{x},t,\boldsymbol{u}))].\end{aligned} \tag{8}$$

Viewing $\boldsymbol{z}$ as a set of latent variables, then $q_\phi(\boldsymbol{z}|\boldsymbol{x},t,\boldsymbol{u})$ is a variational approximation to the true intractable posterior over the latent variables $p_\theta(\boldsymbol{z}|\boldsymbol{x},t,\boldsymbol{u})$. Therefore, if $q_\phi(\boldsymbol{z}|\boldsymbol{x},t,\boldsymbol{u})$ is introduced as an auxiliary inference model associated with the generative model $p_\theta(\boldsymbol{x},t,\boldsymbol{u})$, for which $\boldsymbol{u}=f_\theta(\boldsymbol{x},t,\boldsymbol{z})$ and $\boldsymbol{z}\sim p(\boldsymbol{z})$, then we can use equations (7) and (8) to bound the entropy term in equation (4) as

$$h(p_\theta(\boldsymbol{x},t,\boldsymbol{u}))\geq h(\boldsymbol{z})+\mathbb{E}_{p_\theta(\boldsymbol{x},t,\boldsymbol{u},\boldsymbol{z})}[\log(q_\phi(\boldsymbol{z}|\boldsymbol{x},t,\boldsymbol{u}))]. \tag{9}$$

Note that the inference model $q_\phi(\boldsymbol{z}|\boldsymbol{x},t,\boldsymbol{u})$ plays the role of a variational approximation to the true posterior over the latent variables, and appears naturally using information theoretic arguments in the derivation of the lower bound.

2.3.3. Adversarial training objective

By leveraging the density ratio estimation procedure described in section 2.3.1 and the entropy bound derived in section 2.3.2, we can derive the following loss functions for minimizing the reverse Kullback-Leibler divergence with entropy regularization [35,30]

$$\begin{aligned}\mathcal{L}_\mathcal{D}(\psi)&=\mathbb{E}_{q(\boldsymbol{x},t)p(\boldsymbol{z})}[\log\sigma(T_\psi(\boldsymbol{x},t,f_\theta(\boldsymbol{x},t,\boldsymbol{z})))]+\\&\quad\mathbb{E}_{q(\boldsymbol{x},t,\boldsymbol{u})}[\log(1-\sigma(T_\psi(\boldsymbol{x},t,\boldsymbol{u})))]\end{aligned} \tag{10}$$

$$\mathcal{L}_\mathcal{G}(\theta,\phi)=\mathbb{E}_{q(\boldsymbol{x},t)p(\boldsymbol{z})}[T_\psi(\boldsymbol{x},t,f_\theta(\boldsymbol{x},t,\boldsymbol{z}))+(1-\lambda)\log(q_\phi(\boldsymbol{z}|\boldsymbol{x},t,f_\theta(\boldsymbol{x},t,\boldsymbol{z})))], \tag{11}$$

where $\sigma(x)=1/(1+e^{-x})$ is the logistic sigmoid function. Both loss functions involve expectations over data-points, and we have included subscripts to denote which distributions the expectations are taken with respect to. For example $q(\boldsymbol{x},t)p(\boldsymbol{z})$ indicates that $(\boldsymbol{x},t)$ should be sampled from the empirical data distribution $q(\boldsymbol{x},t)$, while $\boldsymbol{z}$ should be sampled from the prior $p(\boldsymbol{z})$. Equation (10) essentially corresponds to the cross-entropy loss in binary classification and aims to progressively improve the ability of the classifier $T_\psi(\boldsymbol{x},t,\boldsymbol{u})$ to discriminate between "fake" samples $(\boldsymbol{x},t,f_\theta(\boldsymbol{x},t,\boldsymbol{z}))$ produced by the generative model $p_\theta(\boldsymbol{x},t,\boldsymbol{u})$ and "true" samples $(\boldsymbol{x},t,\boldsymbol{u})$ originating from the observed data distribution $q(\boldsymbol{x},t,\boldsymbol{u})$.

Simultaneously, the objective of equation (11) aims at improving the ability of the generator $f_\theta(\boldsymbol{x}, t, \boldsymbol{u})$ to generate increasingly more realistic samples that can "fool" the discriminator $T_\psi(\boldsymbol{x}, t, \boldsymbol{u})$. Specifically, the first term in (11) encourages the joint distribution matching of $p_\theta(\boldsymbol{x}, t, \boldsymbol{u})$ and $q(\boldsymbol{x}, t, \boldsymbol{u})$ via an approximation of the reverse Kullback-Leibler divergence that is computed directly from samples using the density ratio trick outlined in equation (6). Moreover, the encoder $q_\phi(\boldsymbol{z}|\boldsymbol{x}, t, f_\theta(\boldsymbol{x}, t, \boldsymbol{z}))$ distribution appearing in the second term of equation (11) not only serves as an entropic regularization mechanism that allows us to stabilize model training and mitigate the pathology of mode collapse, but also provides a variational approximation to true posterior over the latent variables. This term essentially measures the likelihood of reconstructing the latent variables $\boldsymbol{z}$ that are sampled from the prior $p(\boldsymbol{z})$ given the inputs $(\boldsymbol{x}, t)$ and the generated outputs $f_\theta(\boldsymbol{x}, t, \boldsymbol{z})$, and it defines a loss that is trying to match the latent variables generated by the prior $p(\boldsymbol{z})$ and the latent variables predicted by the encoder with inputs $(\boldsymbol{x}, t, f_\theta(\boldsymbol{x}, t, \boldsymbol{z}))$ using the following procedure. First, latent variables are generated from the prior $p(\boldsymbol{z})$, they are fed through the generator to produce outputs $f_\theta(\boldsymbol{x}, t, \boldsymbol{z})$, which are then fed to the encoder to predict a new set of latent variables. Then the likelihood term $q_\phi(\boldsymbol{z}|\boldsymbol{x}, t, f_\theta(\boldsymbol{x}, t, \boldsymbol{z}))$ essentially tries to match the prior and encoder distributions. This cycling matching process is typically referred to as "cycle-consistency" in the machine learning literature, where such constraints are often enforced in an ad-hoc manner. Interestingly, here this mechanism arises naturally in the derivation of the lower bound for the entropy of the generative model (see equation (8)). The concept of cycle-consistency in generative models has been recently studied in the machine learning community. Several studies [38–40] have demonstrated how cycle-consistency promotes one-to-one nonlinear projections from the physical space (typically high-dimensional) to the latent space (relatively low dimensional), and enhances regularity in the latent space. These properties make such non-linear transformations more robust and "interpretable", as it has been demonstrated that cycle-consistency constraints lead to disentangled latent representations [40].

For the special case of $\lambda = 1.0$, the proposed model resembles a traditional conditional generative adversarial network [41]. Variations of such models are currently appearing in the literature in an ever increasing pace. Here, our main reason for choosing to train such models using the reverse Kullback Leibler divergence is primarily motivated by the introduction of entropy-based regularization that results in a robust and stable training process while helping us to mitigate the issue of mode collapse. Of course, nowadays many adversarial models share the same motivation and other stable formulations exist (e.g. Wasserstein GANs [42,43]). Perhaps a unique advantage of our approach, is that the entropy regularization procedure naturally allows us to compute an approximate posterior distribution over the latent variables. At the prior level where $p(\boldsymbol{z})$ is chosen, the latent variables are not relevant to the physics. However, during model training, our formulation allows us to infer an approximate posterior distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x}, t, \boldsymbol{u})$ which essentially answers the question: which are the most "likely" latent variables, given the system's inputs and outputs? As the relation between inputs and outputs is constrained to approximately satisfy the underlying PDE, the resulting latent variables will also be "informed" by this physical constraint at the posterior level. Overall, these variables aim to summarize the potentially high-dimensional factors that introduce stochasticity in the system. In combination with the favorable properties of cycle-consistency, here we hypothesize that disentangled latent variables can provide a mechanism for obtaining good and physically relevant features for nonlinear model-order reduction.

In theory, the optimal set of parameters $\{\theta^*, \phi^*, \psi^*\}$ correspond to the Nash equilibrium of the two player game defined by the loss functions in equations (10), (11), for which one can show that the exact model distribution and the exact posterior over the latent variables can be recovered [30,44]. In practice, although there is no guarantee that this optimal solution can be attained, the generative model can be trained by alternating between optimizing the two objectives in equations (10), (11) using stochastic gradient descent as

$$\max_{\psi} \mathcal{L}_{\mathcal{D}}(\psi) \tag{12}$$

$$\min_{\theta,\phi} \mathcal{L}_{\mathcal{G}}(\theta, \phi). \tag{13}$$

2.3.4. Adversarial training with physics-informed constrains

In order to learn the physics-informed probabilistic model of equation (3) from data we can extend the adversarial inference framework presented above by appropriately penalizing the loss function of the generator (see equation (11)). The available data correspond to scattered observation pairs $\{(\boldsymbol{x}_i, t_i), \boldsymbol{u}_i\}$, $i = 1, \ldots, N_u$, originating from known initial or boundary conditions, or any other (potentially noisy) measurements of $\boldsymbol{u}(\boldsymbol{x}, t)$. In analogy to the deterministic setting put for in [6] and summarized in section 2.1, by defining $r_\theta(\boldsymbol{x}, t, \boldsymbol{z}) := \frac{\partial}{\partial t} f_\theta(\boldsymbol{x}, t, \boldsymbol{z}) + \mathcal{N}_{\boldsymbol{x}} f_\theta(\boldsymbol{x}, t, \boldsymbol{z})$ we essentially introduce a new conditional probability model $p_\theta(\boldsymbol{r}|\boldsymbol{x}, t, \boldsymbol{z})$ that shares the same parameters as $p_\theta(\boldsymbol{u}|\boldsymbol{x}, t, \boldsymbol{z})$, albeit the underlying neural network that serves as its approximation has different activation functions. However, since we would like to encourage every sample $\boldsymbol{u} = f_\theta(\boldsymbol{x}, t, \boldsymbol{z})$ produced by the generator to satisfy the PDE constraint, we can simply treat the residual as a deterministic variable, i.e., $r_\theta(\boldsymbol{x}, t, \boldsymbol{z}) = r_\theta(\boldsymbol{x}, t)$, and enforce the constraint at a finite set of collocation points N_r by simply minimizing the mean square loss

$$\mathcal{L}_{PDE}(\theta) = \frac{1}{N_r} \sum_{i=1}^{N_r} \| r_\theta(\boldsymbol{x}_i, t_i) - \boldsymbol{r}_i \|^2. \tag{14}$$

Then, the resulting adversarial game for training the physics-informed model of equation (3) takes the form

$$\begin{aligned} &\max_{\psi} \ \mathcal{L}_{\mathcal{D}}(\psi) \\ &\min_{\theta,\phi} \ \mathcal{L}_{\mathcal{G}}(\theta,\phi) + \beta \mathcal{L}_{PDE}(\theta), \end{aligned} \tag{15}$$

where positive values of β can be selected to place more emphasis on penalizing the PDE residual. For $\beta > 0$, the residual loss $\mathcal{L}_{PDE}(\theta)$ acts as a regularization term that approximately enforces the given physical constraint, and, therefore, encourages the generator $p_\theta(\boldsymbol{u}|\boldsymbol{x},t,\boldsymbol{z})$ to produce samples that satisfy the underlying partial differential equation. Also note that this structured approach also encourages the encoder $q_\phi(\boldsymbol{z}|\boldsymbol{x},t,f_\theta(\boldsymbol{x},t,\boldsymbol{z}))$ to learn a set of spatio-temporal latent variables $\boldsymbol{z}$ that are relevant to the underlying physics, possibly opening new directions for probabilistic model order reduction of complex systems.

2.3.5. Predictive distribution

Once the model is trained we can construct a probabilistic ensemble for the solution $p(\boldsymbol{u}|\boldsymbol{x},t,\boldsymbol{z})$ by sampling latent variables from the prior $p(\boldsymbol{z})$ and passing them through the generator to yield samples $\boldsymbol{u} = f_\theta(\boldsymbol{x},t,\boldsymbol{z})$ that are distributed according to the predictive model distribution $p_\theta(\boldsymbol{u}|\boldsymbol{x},t,\boldsymbol{z})$. Note that although the explicit form of this distribution is not known, we can efficiently compute any of its moments via Monte Carlo sampling. The cost of this prediction step is negligible compared to the cost of training the model, as it only involves a single forward pass through the generator function $f_\theta(\boldsymbol{x},t,\boldsymbol{z})$. Typically, we compute the mean and variance of the predictive distribution at a new test point $(\boldsymbol{x}^*,t^*)$ as

$$\mu_{\boldsymbol{u}}(\boldsymbol{x}^*,t^*) = \mathbb{E}_{p_\theta}[\boldsymbol{u}|\boldsymbol{x}^*,t^*,\boldsymbol{z}] \approx \frac{1}{N_s}\sum_{i=1}^{N_s} f_\theta(\boldsymbol{x}^*,t^*,\boldsymbol{z}_i), \tag{16}$$

$$\sigma^2_{\boldsymbol{u}}(\boldsymbol{x}^*,t^*) = \mathbb{V}\mathrm{ar}_{p_\theta}[\boldsymbol{u}|\boldsymbol{x}^*,t^*,\boldsymbol{z}] \approx \frac{1}{N_s}\sum_{i=1}^{N_s} [f_\theta(\boldsymbol{x}^*,t^*,\boldsymbol{z}_i) - \mu_{\boldsymbol{u}}(\boldsymbol{x}^*,t^*)]^2, \tag{17}$$

where $\boldsymbol{z}_i \sim p(\boldsymbol{z})$, $i = 1,\ldots,N_s$, and N_s corresponds to the total number of Monte Carlo samples. In all examples presented in section 3 we have used $N_s = 10^5$ Monte Carlo samples to perform this computation, which can be completed at a fraction of a second as it only involves a forward pass through the trained generator network.

2.3.6. Advantages and caveats of adversarial learning

Since their recent introduction [30,45–47], adversarial learning techniques have provided great flexibility for performing probabilistic computations with arbitrarily complex implicit distributions. Essentially, they have lifted the over-simplified approximations typically used in variational inference (Gaussian approximations, exponential families, etc.) [48], yielding very general and flexible schemes for statistical inference. However, this flexibility comes at a price, as such methods in practice require very careful tuning in order to achieve stable and accurate performance. To this end, recall the training objective defined in equation (15) that introduces an adversarial game between the generator and discriminator networks [30]. In practice, this mini-max optimization problem is solved by alternating stochastic gradient updates between the two competing objectives, and it is highly sensitive on the capacity of the neural networks modeling the generator and discriminator, as well as the relative frequency with which the parameters of each network are updated within each iteration of stochastic gradient descent. To this end, we provide a series of empirical observations and lessons we learned throughout this study that can enhance the robustness and stability of this training procedure:

- Changing the relative number of stochastic gradient updates for the generator K_g and the discriminator K_d is equivalent to changing their neural network architecture. For example, we can reduce the capacity of discriminator by either performing more stochastic gradient updates for the generator, or remove one layer in the neural network architecture of the discriminator.
- Given enough collocation points N_r for penalizing the PDE residual, we can obtain robust uncertainty estimates together with precise predictions simply by tuning the capacity of discriminator and generator networks.
- Typically, by fixing the generator, we expect the discriminator to have some capacity so that the model training dynamics remain stable. But, we do not want the discriminator to be very powerful as in that case there will by very little information from the discriminator that can help the generator to improve towards producing more realistic samples (this a common characteristic of adversarial inference procedures [30]).
- For cases with a small number of training data we should reduce the capacity of the discriminator. This can be achieved by either changing the relative frequency of stochastic gradient updates for the generator and discriminator, or by reducing the capacity of the discriminator neural network architecture.

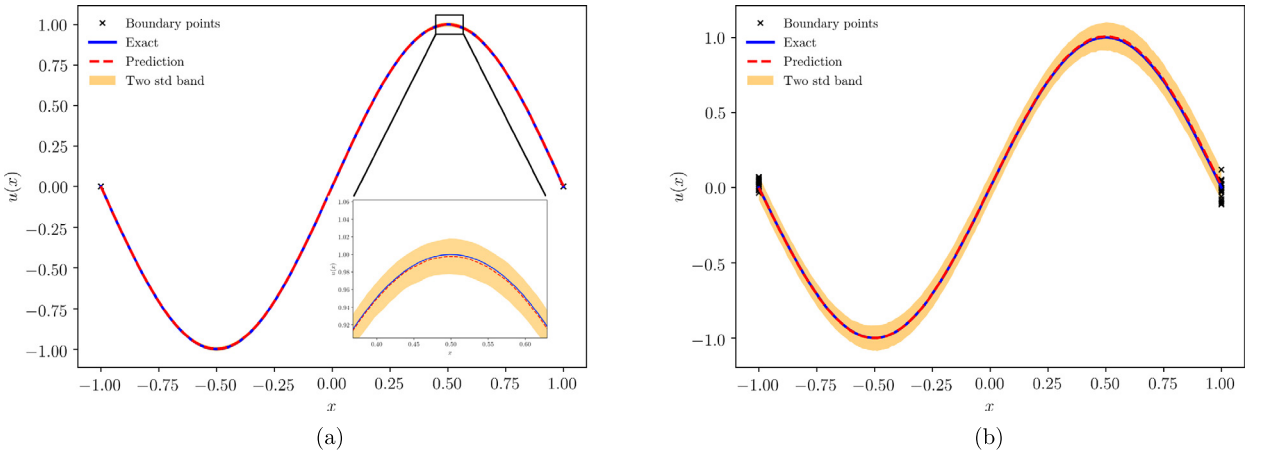

Fig. 1. *A pedagogical example:* (a) Mean and two standard deviations of $p_\theta(u|x,z)$ against the exact solution for deterministic boundary data. (b) Mean and two standard deviations of $p_\theta(u|x,z)$ against the reference Monte Carlo solution for random boundary data corresponding to 5% Gaussian uncorrelated noise. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

3. Results

In all examples we have trained the models for 30,000 stochastic gradient descent steps using the Adam optimizer [49] with a learning rate of 10^{-4}, while fixing a one-to-five ratio for the discriminator versus generator updates. Moreover, we have fixed the entropic regularization and the residual penalty parameters to $\lambda = 1.5$ and $\beta = 1.0$, respectively. The proposed algorithms were implemented in Tensorflow v1.10 [33], and computations were performed in single precision arithmetic on a single NVIDIA Tesla P100 GPU card. All data and code accompanying this manuscript will be made available at https://github.com/PredictiveIntelligenceLab/UQPINNs.

3.1. A pedagogical example

Let us illustrate the basic capabilities of the proposed methods through a simple example corresponding to the following nonlinear second-order ordinary differential equation

$$\begin{aligned} &u_{xx} - u^2 u_x = f(x), \qquad x \in [-1,1], \\ &f(x) = -\pi^2 \sin(\pi x) - \pi \cos(\pi x) \sin^2(\pi x), \end{aligned} \tag{18}$$

subject to random boundary conditions $u(-1), u(1) \sim \mathcal{N}(\mathbf{0}, \sigma_n^2 \boldsymbol{I})$. For this simple example, the deterministic solution corresponding to $\sigma_n^2 = 0$ can be readily obtained as $u(x) = \sin(\pi x)$. Given N_u observations of $u(x)$ corresponding to different realizations of the random boundary conditions our goal is to obtain a probabilistic representation of the solution $p_\theta(u|x,\boldsymbol{z})$ by training a physics-informed generative model of the form $u = f_\theta(x,\boldsymbol{z})$, $\boldsymbol{z} \sim p(\boldsymbol{z})$ that is constrained by equation (18). To this end, we introduce three deterministic mappings parametrized by deep neural networks, namely $f_\theta(x,\boldsymbol{z})$, $q_\phi(x,u)$, and $T_\psi(x,u)$ corresponding to the generator, encoder, and discriminator functions introduced in section 2.3. By construction, we also obtain a physics-informed neural network $r_\theta(x)$ corresponding to the deterministic residual of equation (18) that will be used to approximately enforce the differential equation constraint at a set of $N_r = 100$ randomly distributed collocation points $x \in [-1,1]$. All neural networks were chosen to have two hidden layers with 50 neurons in each layer, and a hyperbolic tangent activation function. Moreover, the dimensionality of the latent variables was set to one, i.e. $\boldsymbol{z} = z$, and we have assumed an isotropic standard normal prior, namely $p(z) \sim \mathcal{N}(0,1)$. As the training data for $u(x)$ reflects the uncertainty in the boundary conditions, the role of the latent variables z is to enable the propagation of this uncertainty into the predicted solution obtained through the generative model $p_\theta(u|x,z)$.

Here we have considered two cases corresponding to deterministic and random boundary conditions, namely (i) $\sigma_n^2 = 0$ (i.e., noise-free data), and (ii) $\sigma_n^2 = 0.05$ (i.e., 5% Gaussian uncorrelated noise). In all cases, the training data consists of $N_u = 20$ realizations for each boundary point, $u(-1), u(1)$, and a total of $N_r = 100$ collocation points for enforcing the residual of equation (18). Our probabilistic predictions for this example are summarized in Figs. 1 and 2. Specifically, Fig. 1(a) shows the generative model predictive mean and two standard deviations, plotted against the exact solution of this problem. Note that this case corresponds to deterministic training data for the boundary conditions, hence the exact solution is deterministic, and the prediction error here is measured as $\mathcal{E}_{\mathbb{L}_2} = 1.36 \cdot 10^{-3}$ in the relative $\mathbb{L}_2$ norm

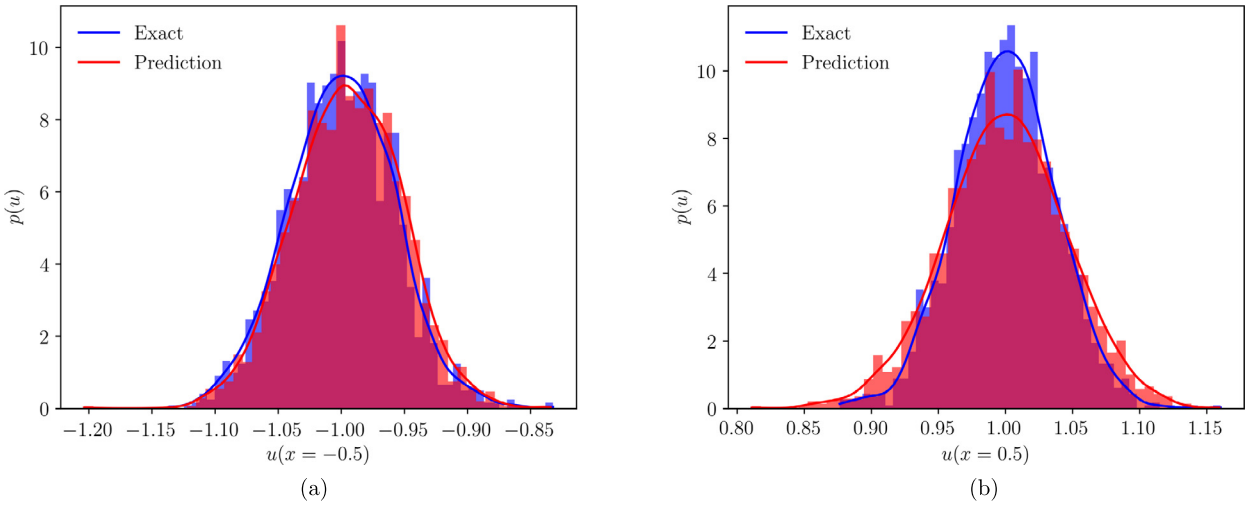

Fig. 2. *A pedagogical example:* Predicted marginal densities against the reference Monte Carlo solution. (a) $p_\theta(u|x=-0.5,z)$. (b) $p_\theta(u|x=+0.5,z)$.

$$\mathcal{E}_{\mathbb{L}_2} := \frac{\sqrt{\sum_{i=1}^{N^*}[\mu_{\boldsymbol{u}}(x_i^*) - u(x_i^*)]^2}}{\sqrt{\sum_{i=1}^{N^*}[u(x_i^*)]^2}}, \tag{19}$$

where $N^* = 200$ denotes the total number of equidistant test points x^* in the interval $[-1, 1]$. Moreover, the variance shown in the inset of Fig. 1(a) quantifies the uncertainty associated with the generative model predictions. As our method is based on the assumption of finite number of accessible data, this uncertainty still exists when there is no noise on the boundary is because of the finite number of data used for training. To this end, we have empirically observed that as the number of training data on the boundary is increased, the predicted uncertainty becomes much smaller.

Fig. 1(b) shows the resulting prediction and uncertainty estimates corresponding to random boundary conditions, compared against a reference mean solution obtained numerically using a spectral method with 2,000 Monte Carlo samples. In this case, the predictive uncertainty of the generative model reflects the aggregate *total* uncertainty due to both randomness in the boundary conditions and the inherent epistemic uncertainty in the neural network approximation. In other words, it is not evident which part of the resulting uncertainty should be attributed to the finite size of the training data versus the system stochasticity/noise. This is attributed to our assumed representation of $\boldsymbol{u} = f_\theta(\boldsymbol{x}, t, \boldsymbol{z})$, in which the latent variables are entangled with the inputs in a complex nonlinear way. On one hand, this allows us to tackle cases involving complex noise processes (see for e.g. the results presented in the next section), but it prevents us to directly quantify the influence of the noise versus the finite number of training data on the resulting predicted uncertainty.

Finally, as the generative model can return a complete statistical characterization of the solution by means of its conditional probability density $p_\theta(u|x,z)$, in Fig. 2 we provide a visual comparison of the one-dimensional marginals between our predictions and the reference Monte Carlo solution corresponding to the spatial locations $x = -0.5$ and $x = 0.5$.

In order to highlight the sensitivity of our results with respect to the entropy and the PDE regularization penalty parameters, λ and β, respectively, we have performed additional numerical studies to highlight the sensitivity of our results on the parameters λ and β. Specifically, we have repeated our simulations for different values of λ and β and computed the reverse Kullback-Leibler divergence of the predicted distribution $p_\theta(u|x,z)$ against the reference "true" distribution $p_e(u|x)$ computed by Monte Carlo sampling of a spectral solver. The reverse KL-divergence is then averaged in a uniform discretization of the interval $[-1, 1]$, i.e. $x \sim p(x) = U[-1, 1]$ as

$$\mathbb{E}_{p(x)p(z)}\{\mathbb{KL}[p_\theta(u|x,z)||p_e(u|x)]\}. \tag{20}$$

The results of this experiment are summarized in Table 1 and reveal the mode-collapse pathology of generative adversarial models for $\lambda = 1.0$, as well as the merits of incorporating the physics-informed constraint for $\beta > 0$. Overall, these results indicate that for $\lambda > 1.0$ and $\beta > 0$ the proposed methodology returns robust results that exhibit low sensitivity on the respective values of λ and β.

Albeit simple, this example aims to demonstrate the basic capabilities of the proposed methodology in propagating uncertainty through non-linear partial differential equations. In contrast to previous approaches to inferring solutions of partial differential equations from data [50–53], the proposed methodology does not rely on Gaussian assumptions, and it can directly tackle nonlinear problems without any need for linearization.

Table 1
A pedagogical example: Average reverse KL-divergence between the predicted data and the ground truth with parameters λ and β for the example presented in section 3.1 of the manuscript.

β \ λ	1.0	1.5	2.0	5.0
0	5.0e+05	7.5e+01	6.0e+01	4.4e+01
1.0	3.3e+02	1.8e−01	2.9e−01	2.0e−01
2.0	2.1e+02	1.7e−01	5.0e−02	1.2e−01
5.0	3.5e+01	1.8e−01	1.9e−01	1.1e−01

3.2. Burgers equation

In this example we aim to provide a comprehensive systematic study to quantify the robustness of the proposed methods with respect to different parameter choices. We will do so through the lens of a more challenging canonical problem involving the non-linear time-dependent Burgers equation in one spatial dimension:

$$\begin{aligned} &u_t + uu_x - \nu u_{xx} = 0, \qquad x \in [-1, 1], t \in [0, 1], \\ &u(0, x) = -\sin(\pi x), \\ &u(t, -1) = u(t, 1) = 0, \end{aligned} \tag{21}$$

where the viscosity parameter is chosen as $\nu = (0.01/\pi)$ in order to generate a strongly nonlinear response that leads to the development of a steep internal layer centered at $x = 0.0$. This is one of the few nonlinear partial differential equations that admits an exact solution through the Cole-Hopf transformation [54]; a solution that will be subsequently used to test the validity of our predictions.

Here we represent the unknown solution $u(x, t)$ using a physics-informed generative model of the form $u = f_\theta(x, t, \boldsymbol{z})$, and we will introduce parametric functions corresponding to a generator $f_\theta(x, t, \boldsymbol{z})$, an encoder $q_\phi(x, t, u)$, and a discriminator $T_\psi(x, t, u)$ all constructed using deep feed-forward neural networks. The baseline architectures for the generator and the encoder have 4 hidden layers with 50 neurons per layers, while the discriminator network has 3 hidden layers and 50 neurons per layer. The activation function in all cases is chosen to be a hyperbolic tangent non-linearity. The prior over the latent variables $p(\boldsymbol{z})$ is chosen again to be a one-dimensional isotropic Gaussian distribution, i.e. $\boldsymbol{z} = z$, $z \sim \mathcal{N}(0, 1)$.

First we consider a baseline scenario, in which we train our probabilistic model using a data-set comprising of $N_u = 150$ noisy-free input/output pairs for $u(x, t)$ – 50 points for the initial condition (see Fig. 3(a)) and 50 points for each of the domain boundaries – plus an additional $N_r = 10{,}000$ collocation points for enforcing the residual of the Burgers equation using the loss of equation (14). All data points were randomly selected within the bounds given in equation (21). The result of this experiment is summarized in 4 where we report the predicted mean solution, as well as the uncertainty associated with this prediction as quantified by two standard deviations of the generative model $p_\theta(u|x, t, z)$. As the training data for this case is noise-free, the solution to this problem is deterministic, and the resulting uncertainty captured in $p_\theta(u|x, t, z)$ provides a quantification of the neural network approximation error due to the finite number of training data, which is measured as $\mathcal{E}_{\mathbb{L}_2} = 4.1 \cdot 10^{-2}$ in the relative $\mathbb{L}_2$ norm. As discussed in [31], a higher approximation accuracy can be achieved by training the generative model using a quasi-Newton optimizer (e.g. L-BFGS [55]), however here we chose to use stochastic gradient descent using Adam updates [49] in order to highlight the ability of the proposed method to return uncertainty estimates when the model predictions are not perfectly accurate.

Second, we repeat the same test for a more complicated scenario in which the initial condition has been corrupted by non-additive, non-Gaussian noise as shown in Fig. 3(b), where the noise variance is larger around $x = 0.0$, therefore amplifying the effect of uncertainty on the steep internal layer formation. Here the neural network architecture as well as the number and location of training points have been kept fixed as described above, but the initial condition is now corrupted as

$$u(x, 0) = -\sin(\pi(x + 2\delta)) + \delta, \qquad \delta = \frac{\epsilon}{\exp(3|x|)}, \qquad \epsilon \sim N(0, 0.1^2). \tag{22}$$

The results of this experiment are summarized in Fig. 5. We observe that the resulting generative model $p_\theta(u|x, t, z)$ can effectively capture the uncertainty in the resulting spatio-temporal solution due to the propagation of the input noise process through the complex non-linear dynamics of the Burgers equation. As expected, the uncertainty concentrates around the steep gradients region at $x = 0.0$. Although we only plot the first two moments of the solution, we must emphasize that the generative model $p_\theta(u|x, t, z)$ provides a complete probabilistic characterization of its non-Gaussian statistics.

In order to further investigate the performance of the proposed methodology for different parameter settings, we have performed a series of comprehensive systematic studies that aim to quantify the sensitivity of the resulting predictions on: (i) the neural network initialization, (ii) the total number of training and collocation points, (iii) the neural network architecture, and (iv) the adversarial training procedure. The results of these systematic studies are provided in Appendix A.

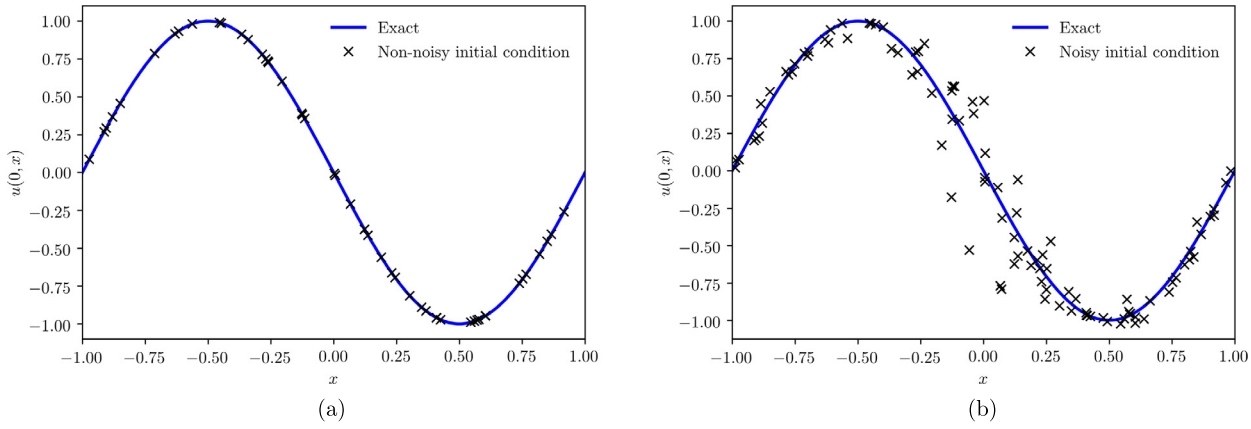

Fig. 3. *Burgers equation:* (a) Exact initial condition and noise-free training data (50 points). (b) Training data corresponding to a single realization of the non-additive noise corruption process (100 points, generated by equation (22)).

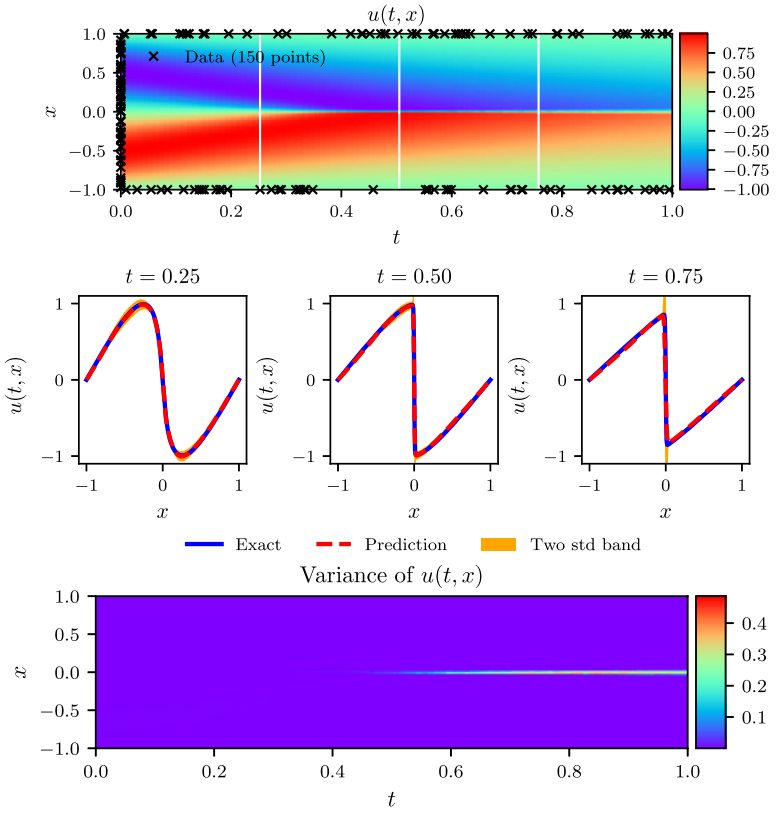

Fig. 4. *Burgers equation with noise-free data: Top:* Mean of $p_\theta(u|x,t,z)$, along with the location of the noisy training data $\{(x_i,t_i),u_i\}$, $i=1,\ldots,N_u$. *Middle:* Prediction and predictive uncertainty at $t=0.25$, $t=0.5$ and $t=0.75$. *Bottom:* Variance of $p_\theta(u|x,t,z)$.

3.3. Discovery of constitutive laws for flow through porous media

In our final example, we aim to demonstrate the ability of the proposed methods to discover unknown constitutive relationships directly from data with quantified uncertainty. To this end, we revisit the Darcy flow example put forth in [56] corresponding to a two-dimensional nonlinear diffusion equation with an unknown state-dependent diffusion coefficient

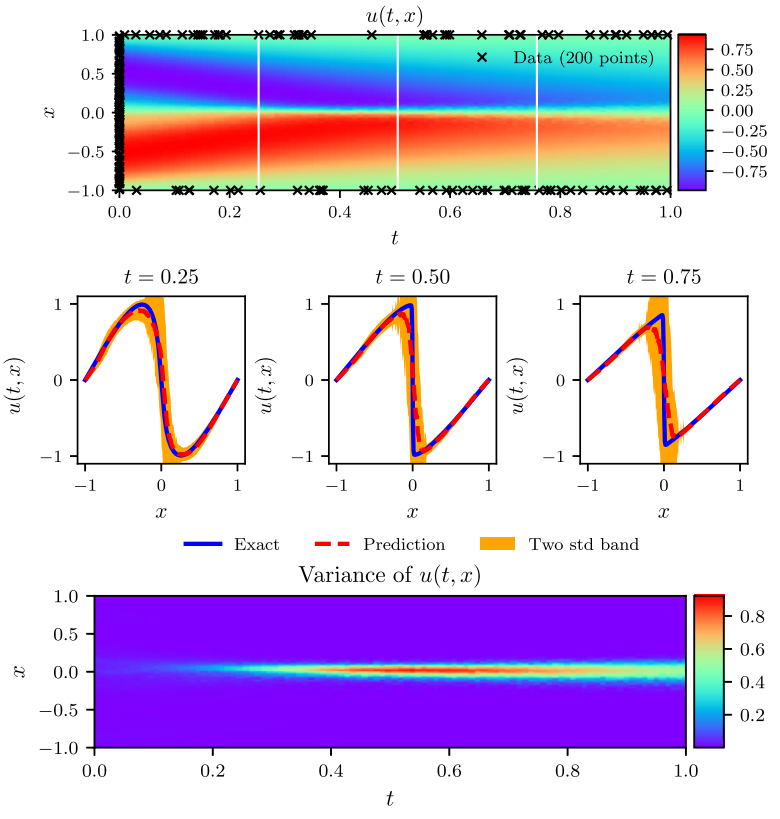

Fig. 5. *Burgers equation with noisy data: Top:* Mean of $p_\theta(u|x,t,z)$, along with the location of the training data $\{(x_i,t_i),u_i\}$, $i=1,\ldots,N_u$. *Middle:* Prediction and predictive uncertainty at $t=0.25$, $t=0.5$ and $t=0.75$. *Bottom:* Variance of $p_\theta(u|x,t,z)$.

$$\begin{aligned} \nabla_{\boldsymbol{x}}\cdot[k(u)\nabla_{\boldsymbol{x}}u(\boldsymbol{x})] &= 0, \qquad && \boldsymbol{x}=(x_1,x_2)\in\Omega=(0,L_1)\times(0,L_2) \\ u(\boldsymbol{x}) &= u_0, && x_1=L_1 \\ -k(u)\frac{\partial u(\boldsymbol{x})}{\partial x_1} &= q, && x_1=0 \\ \frac{\partial u(\boldsymbol{x})}{\partial x_2} &= 0, && x_2=\{0,L_2\}, \end{aligned} \tag{23}$$

where $q=8.25\times10^{-5}$ m/s and $u_0=-10$ m are known boundary conditions. In order to benchmark and validate our model predictions we consider a realistic data-set generated using the Subsurface Transport Over Multiple Phases (STOMP) code [57] with the van Genuchten model [58] for $k(u)$ which reads as

$$\begin{aligned} k(s(u)) &= K_s s^{\frac{1}{2}}[1-(1-s^{\frac{1}{m}})^m]^2 \\ s(u) &= \{1+[\alpha(u_g-u)]^{\frac{1}{1-m}}\}^{-m}, \end{aligned} \tag{24}$$

with the following parameter values: $K_s=8.25\times10^{-4}$ m/s, $u_g=0$, $m=0.469$, $\alpha=0.1$, $L_1=10$ m and $L_2=10$ m.

Our goal is twofold: we aim to construct a physics-informed probabilistic model for $p_\theta(u|\boldsymbol{x},\boldsymbol{z})$, and simultaneously learn the unknown state-dependent diffusion coefficient $k(u)$ directly from data on $u(\boldsymbol{x})$ (i.e., we assume no measurements of $k(u)$). To this end, in addition to the three deterministic mappings $f_\theta(\boldsymbol{x},\boldsymbol{z})$, $q_\phi(\boldsymbol{x},u)$, and $T_\psi(\boldsymbol{x},u)$ corresponding to the generator, encoder, and discriminator described in section 2.3, here we also introduce another neural network $f_\gamma(u)$ for approximating $k(u)$. The parameters of $f_\gamma(u)$ are essentially inherited by the physics-informed residual network $r_{\theta,\gamma}(\boldsymbol{x}):=\nabla_{\boldsymbol{x}}\cdot[f_\gamma(f_\theta(\boldsymbol{x},\boldsymbol{z}))\nabla_{\boldsymbol{x}}f_\theta(\boldsymbol{x},\boldsymbol{z}))]$ that aims to enforce the residual of equation (23) at the N_r collocation points for any set of latent variables $\boldsymbol{z}$. All neural networks are chosen to have 2 hidden layers with 50 neurons per each, and a hyperbolic tangent activation function, while the probabilistic model for $p_\theta(u|\boldsymbol{x},\boldsymbol{z})$ assumes a two dimensional latent space with an isotropic Gaussian prior, i.e., $\boldsymbol{z}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})$.

By construction, our probabilistic model for $p_\theta(u|\boldsymbol{x},\boldsymbol{z})$ can return predictions of the unknown solution $u(\boldsymbol{x})$ with quantified uncertainty. We can then use this model to propagate uncertainty in our predictions of $k(u)$ via Monte Carlo sampling. Specifically, once the model is trained end-to-end, we can easily generate samples of $u(\boldsymbol{x})$ from $p_\theta(u|\boldsymbol{x},\boldsymbol{z})$ and propagate

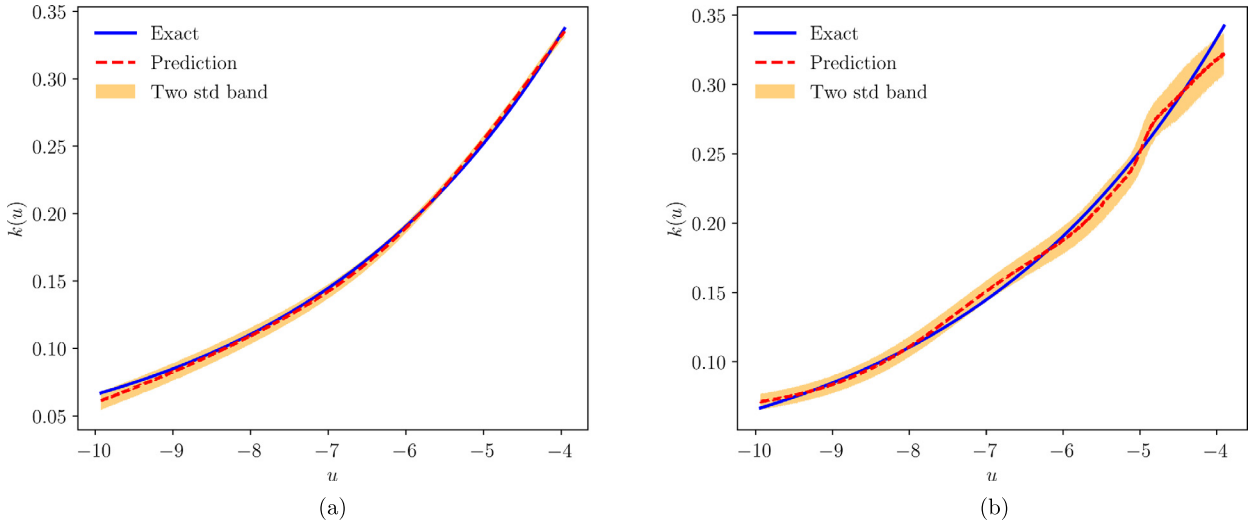

Fig. 6. Prediction with quantified uncertainty of unknown state-dependent diffusion coefficient $k(u)$ compared against the reference solution obtained from the Subsurface Transport Over Multiple Phases (STOMP) code [57] with the van Genuchten model [58] (see equation (24)). (a) Noise-free training data for $u(\boldsymbol{x})$. (b) Noisy training data for $u(\boldsymbol{x})$ with noise level of 5%.

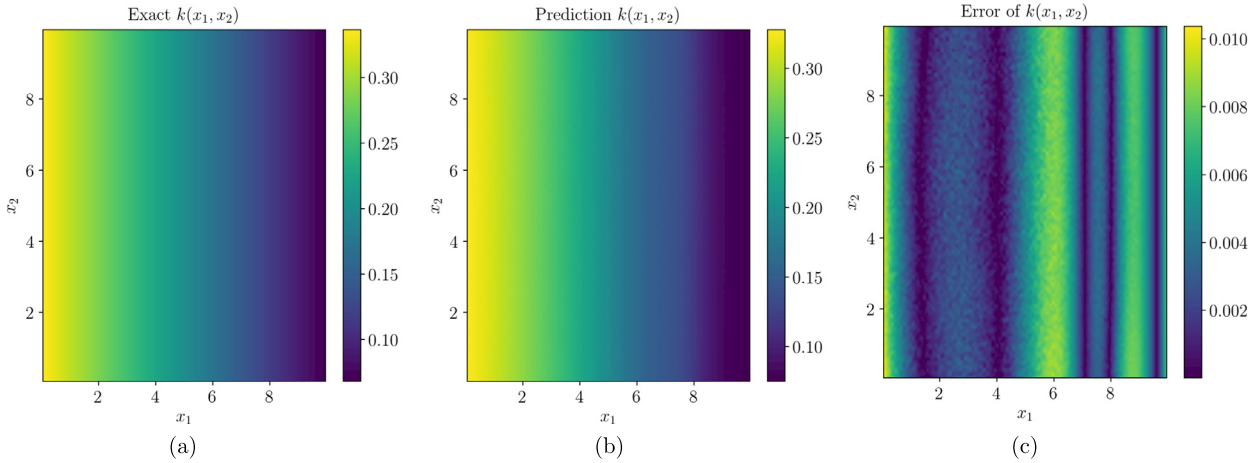

Fig. 7. Prediction of unknown state-dependent diffusion coefficient $K(u(x_1, x_2))$. (a) Reference solution obtained from the Subsurface Transport Over Multiple Phases (STOMP) code [57] with the van Genuchten model [58] (see equation (24)). (b) Predictive mean of the generative model $p_{\theta,\gamma}(k|\boldsymbol{x}, \boldsymbol{z})$ trained on noise-free data for $u(\boldsymbol{x})$. (c) Absolute point-wise prediction error.

them through $f_\gamma(u)$ to obtain a samples for $k(u)$. Essentially this results in an implicit generative model $p_{\theta,\gamma}(k|\boldsymbol{x}, \boldsymbol{z})$ which can fully characterize uncertainty in our predictions of the unknown state-dependent diffusion coefficient.

Here we also have considered two cases corresponding to noise-free training data and noisy data corrupted by 5% Gaussian uncorrelated noise. In the noise-free case we used $N_u = 600$ scattered measurements of the unknown solution $u(\boldsymbol{x})$ – 200 inside the domain Ω and 100 on each one of the four boundaries – and total number of $N_r = 10,000$ randomly selected collocation points inside the domain for penalizing the residual of equation (23). For the noisy case we chose $N_u = 1{,}400$ scattered measurements of the unknown solution $u(\boldsymbol{x})$ – 1,000 inside the domain Ω and 100 on each one of the four boundaries – while still keeping $N_r = 10,000$ collocation points. Fig. 6 summarizes the results for both cases by showing the predictive mean and two standard deviations of the corresponding generative model $p_{\theta,\gamma}(k|\boldsymbol{x}, \boldsymbol{z})$, against the reference (deterministic) solution obtained from the Subsurface Transport Over Multiple Phases (STOMP) code [57] with the van Genuchten model [58] (see equation (24)). Evidently, the generative model is able to recover a sensible prediction for the unknown state-dependent diffusion coefficient with quantitative uncertainty, even when the training data on $u(\boldsymbol{x})$ is corrupted by noise. Moreover, notice that $k(u)$ implicitly depends on the spatial coordinates $\boldsymbol{x} = (x_1, x_2)$. In Fig. 7 we present the resulting prediction for $K(u(x_1, x_2))$ corresponding to the noise-free case, against the reference solution, as well as their point-wise absolute error.

Table A.2
Relative $\mathbb{L}_2$ prediction error for different neural network initializations using a randomized seed.

Relative $\mathcal{L}_2$ error				
4.1e−02	7.9e−02	4.4e−02	4.0e−02	3.8e−02
3.2e−02	5.7e−02	4.7e−02	6.5e−02	4.0e−02
3.5e−02	3.5e−02	6.4e−02	4.0e−02	4.9e−02

Again we must emphasize that these predictions are obtained without ever observing any data on $k(u)$, while they are accompanied by quantitative estimates that jointly characterize the uncertainty due to noise in the training data for $u(\boldsymbol{x})$, and the underlying approximation error of the neural networks. This theme of consistently inferring correlated continuous quantities of interest from a small set of measurements by leveraging the underlying laws of physics is a great example of the exciting capabilities that physics informed machine learning has to offer.

4. Conclusions

We presented a class of probabilistic physics-informed neural networks that are capable of approximating arbitrary conditional probability densities, while being constrained to generate samples that approximately satisfy given partial differential equations. Moreover, we have derived a flexible regularized adversarial inference framework that enables the end-to-end training of such models directly from noisy and incomplete measurements. Uncertainty in the system inputs and/or outputs is captured through a set of latent variables that are relevant to the underlying physics, and could possibly open new directions for probabilistic model-order reduction of complex systems. These developments allow us to perform probabilistic computations for uncertain systems, train deep generative models in small data regimes, handle complex noise processes, and seamlessly carry out uncertainty propagation studies for physical systems without the need for repeated evaluation of experiments and numerical simulations.

Although the proposed adversarial inference framework provides great flexibility for performing probabilistic computations and approximating arbitrarily complex and high-dimensional probability distributions, it relies on carefully tuning the interplay between the generator and discriminator networks. This is a known limitation of adversarial algorithms, and, although several works have led to improvements [59,42], it still largely remains an open research problem. An alternative path for enhancing the robustness of the inference procedure, while not compromising its ability to handle complex probability distributions, comes through the use of invertible transformations and flow-based generative models [60,61]. Future work will examine the applications of such models in the context of physics-informed neural networks with the goal of enhancing the robustness of the proposed methods and scaling them to more realistic systems.

Acknowledgements

This work received support from the US Department of Energy under the Advanced Scientific Computing Research program (grant DE-SC0019116) and the Defense Advanced Research Projects Agency under the Physics of Artificial Intelligence program (grant HR00111890034). We would also like to thank Dr. Alexandre Tartakovsky from the Pacific Nortwest National Laboratory for providing the Darcy flow data-set.

Appendix A. Sensitivity studies

Here we provide results on a series of comprehensive systematic studies that aim to quantify the sensitivity of the resulting predictions on: (i) the neural network initialization, (ii) the total number of training and collocation points, (iii) the neural network architecture, and (iv) the adversarial training procedure. In all cases we have used the non-linear Burgers defined in section 3.2 as a prototype problem.

A.1. Sensitivity with respect to the neural network initialization

In order to quantify the sensitivity of the proposed methods with respect to the initialization of the neural networks, we have considered a noise-free data set comprising of $N_u = 150$ and $N_r = 10000$ training and collocation points, respectively, and fixed the architecture for generator neural networks to include 4 hidden layers with 50 neurons each and discriminator neural networks to include 3 hidden layers with 50 neurons each, and a hyperbolic tangent activation function. Then we have trained an ensemble of 15 cases all starting from a normal Xavier initialization [62] for all network weights (with a randomized seed), and a zero initialization for all bias parameters. In Table A.2 we report the relative error between the predicted mean solution and the known exact solution for this problem for all 15 randomized trials using at set of 25600 randomly selected test points. Evidently, our results are robust with respect to the neural network initialization as in all cases the stochastic gradient descent training procedure converged roughly to the same solution. We can summarize this result by reporting the mean and the standard deviation of the relative $\mathbb{L}_2$ error as

$$\hat{\mathcal{L}_2} \in [\mu_L - \sigma_L, \mu_L + \sigma_L] = [4.7 \times 10^{-2} - 1.3 \times 10^{-2}, 4.7 \times 10^{-2} + 1.3 \times 10^{-2}].$$

Table A.3
Relative $\mathbb{L}_2$ prediction error for different number of training and collocation points N_u and N_r, respectively.

N_u \ N_r	10	100	250	500	1000	5000	10000
60	9.3e−01	5.6e−01	4.8e−01	5.0e−02	1.9e−01	5.0e−02	5.1e−02
90	5.8e−01	5.3e−01	3.5e−01	1.5e−01	4.9e−02	1.0e−01	5.8e−02
150	6.7e−01	1.4e−01	3.0e−01	3.6e−02	4.9e−02	1.2e−01	4.7e−02

Table A.4
Relative $\mathbb{L}_2$ prediction error for different feed-forward architectures for the generator, encoder, and the discriminator. The total number of layers of the latter was always chosen to be one less than the number of layers for generator.

N_g \ N_n	20	50	100
2	4.2e−01	3.8e−01	5.7e−01
3	6.5e−02	3.5e−02	2.1e−02
4	9.3e−02	4.7e−02	5.4e−02

Table A.5
Relative $\mathbb{L}_2$ error with different number of training for generator and discriminator in each epoch.

K_g \ K_d	1	2	5
1	3.5e−01	5.0e−01	1.5e+00
2	4.3e−02	3.2e−01	5.4e−01
5	4.7e−02	2.3e−01	7.0e−01

A.2. Sensitivity with respect to the total number of training and collocation points

In this study our goal is to quantify the sensitivity of our predictions with respect to the total number of training and collocation points N_u and N_r, respectively. As before, we have considered noise-free data sets, and fixed the architecture for generator neural networks to include 4 hidden layers with 50 neurons each and discriminator neural networks to include 3 hidden layers with 50 neurons each, a hyperbolic tangent activation function, and a normal Xavier initialization [62] for all network weights and zero initialization for all network biases. The results of this study are summarized in Table A.3, indicating that as the number of collocation points are increased, a more accurate prediction is obtained. This observation is in agreement with the original results of Raissi et al. [6,7] for deterministic physics-informed neural networks, indicating the role of the residual loss as an effective regularization mechanism for training deep generative models in small data regimes.

A.3. Sensitivity with respect to the neural network architecture

In this study we aim to quantify the sensitivity of our predictions with respect to the architecture of the neural networks that parametrize the generator, the discriminator, and the encoder. Here we have fixed the number of noise-free training data to $N_u = 150$ and $N_r = 10000$, and we kept the number of layers for discriminator to always be one less than the number of layers for generator (e.g., if the number of layers for generator is two then the number of layers for discriminator is one, etc.). In all cases, we have used a hyperbolic tangent non-linearity and a normal Xavier initialization [62]. In Table A.4 we report the relative $\mathbb{L}_2$ prediction error for different feed-forward architectures for the generator, discriminator, and encoder (i.e., different number of layers and number of nodes in each layer). The general trend suggests that as the neural network capacity is increased we obtain more accurate predictions, indicating that our physics-informed constraint on the PDE residual can effectively regularize the training process and safe-guard against over-fitting. We note number of neurons in each layer as N_n and number of layers for generator (encoder) as N_g.

A.4. Sensitivity with respect to the adversarial training procedure

Finally, we test the sensitivity with respect to the adversarial training process. To this end, we have fixed the number of noise-free training data to $N_u = 150$ and $N_r = 10000$, and the neural network architecture to be the same as A.2, and we vary the total number of training steps for the generator K_g and the discriminator K_d within each stochastic gradient descent iteration. The results of this study are presented in Table A.5 where we report the relative $\mathbb{L}_2$ prediction error. These results reveal the high sensitivity of the training dynamics on the interplay between the generator and discriminator

networks, and pinpoint on the well known peculiarity of adversarial inference procedures which require a careful tuning of K_g and K_d for achieving stable performance in practice.

References

[1] 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.
[2] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444.
[3] B.M. Lake, R. Salakhutdinov, J.B. Tenenbaum, Human-level concept learning through probabilistic program induction, Science 350 (2015) 1332–1338.
[4] B. Alipanahi, A. Delong, M.T. Weirauch, B.J. Frey, Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning, Nature Biotechnology 33 (2015) 831–838.
[5] I. Goodfellow, Y. Bengio, A. Courville, Y. Bengio, Deep Learning, vol. 1, MIT Press, Cambridge, 2016.
[6] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics informed deep learning (part i): data-driven solutions of nonlinear partial differential equations, preprint, arXiv:1711.10561, 2017.
[7] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics informed deep learning (part ii): data-driven discovery of nonlinear partial differential equations, preprint, arXiv:1711.10566, 2017.
[8] M. Raissi, P. Perdikaris, G.E. Karniadakis, Multistep neural networks for data-driven discovery of nonlinear dynamical systems, preprint, arXiv:1801.01236, 2018.
[9] M. Raissi, G.E. Karniadakis, Hidden physics models: machine learning of nonlinear partial differential equations, J. Comput. Phys. 357 (2018) 125–141.
[10] M. Raissi, Deep hidden physics models: deep learning of nonlinear partial differential equations, preprint, arXiv:1801.06637, 2018.
[11] D.C. Psichogios, L.H. Ungar, A hybrid neural network-first principles approach to process modeling, AIChE J. 38 (1992) 1499–1511.
[12] I.E. Lagaris, A. Likas, D.I. Fotiadis, Artificial neural networks for solving ordinary and partial differential equations, IEEE Trans. Neural Netw. 9 (1998) 987–1000.
[13] P. Perdikaris, L. Grinberg, G.E. Karniadakis, Multiscale modeling and simulation of brain blood flow, Phys. Fluids 28 (2016) 021304.
[14] D. Rossinelli, Y.-H. Tang, K. Lykov, D. Alexeev, M. Bernaschi, P. Hadjidoukas, M. Bisson, W. Joubert, C. Conti, G. Karniadakis, et al., The in-silico lab-on-a-chip: petascale and high-throughput simulations of microfluidics at cell resolution, in: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, ACM, 2015, p. 2.
[15] J. Šukys, U. Rasthofer, F. Wermelinger, P. Hadjidoukas, P. Koumoutsakos, Optimal fidelity multi-level Monte Carlo for quantification of uncertainty in simulations of cloud cavitation collapse, preprint, arXiv:1705.04374, 2017.
[16] J.T. Oden, R. Moser, O. Ghattas, Computer predictions with quantified uncertainty, part II, SIAM News 43 (2010) 1–4.
[17] R. Ghanem, P.D. Spanos, Polynomial chaos in stochastic finite elements, J. Appl. Mech. 57 (1990) 197–202.
[18] D. Xiu, G.E. Karniadakis, The Wiener–Askey polynomial chaos for stochastic differential equations, SIAM J. Sci. Comput. 24 (2002) 619–644.
[19] H.N. Najm, Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics, Annu. Rev. Fluid Mech. 41 (2009) 35–52.
[20] T. Gerstner, M. Griebel, Numerical integration using sparse grids, Numer. Algorithms 18 (1998) 209.
[21] M. Eldred, J. Burkardt, Comparison of non-intrusive polynomial chaos and stochastic collocation methods for uncertainty quantification, in: 47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition, 2009, p. 976.
[22] A. Barth, C. Schwab, N. Zollinger, Multi-level Monte Carlo finite element method for elliptic PDEs with stochastic coefficients, Numer. Math. 119 (2011) 123–161.
[23] B. Peherstorfer, K. Willcox, M. Gunzburger, Optimal model management for multifidelity Monte Carlo estimation, SIAM J. Sci. Comput. 38 (2016) A3163–A3194.
[24] G. Berkooz, P. Holmes, J.L. Lumley, The proper orthogonal decomposition in the analysis of turbulent flows, Annu. Rev. Fluid Mech. 25 (1993) 539–575.
[25] O. Le Maître, O.M. Knio, Spectral Methods for Uncertainty Quantification: With Applications to Computational Fluid Dynamics, Springer Science & Business Media, 2010.
[26] I. Bilionis, N. Zabaras, Multi-output local Gaussian process regression: applications to uncertainty quantification, J. Comput. Phys. 231 (2012) 5718–5746.
[27] 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.
[28] P. Perdikaris, D. Venturi, G.E. Karniadakis, Multifidelity information fusion algorithms for high-dimensional systems and massive data sets, SIAM J. Sci. Comput. 38 (2016) B521–B538.
[29] D.P. Kingma, M. Welling, Auto-encoding variational Bayes, preprint, arXiv:1312.6114, 2013.
[30] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in Neural Information Processing Systems, 2014, pp. 2672–2680.
[31] 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.
[32] A.G. Baydin, B.A. Pearlmutter, A.A. Radul, J.M. Siskind, Automatic differentiation in machine learning: a survey, preprint, arXiv:1502.05767, 2015.
[33] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al., Tensorflow: a system for large-scale machine learning, in: OSDI, vol. 16, 2016, pp. 265–283.
[34] B. Shahriari, K. Swersky, Z. Wang, R.P. Adams, N. De Freitas, Taking the human out of the loop: a review of Bayesian optimization, Proc. IEEE 104 (2016) 148–175.
[35] C. Li, J. Li, G. Wang, L. Carin, 2018, Learning to sample with adversarially learned likelihood-ratio.
[36] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, Improved techniques for training GANS, in: Advances in Neural Information Processing Systems, 2016, pp. 2234–2242.
[37] H. Akaike, Information theory and an extension of the maximum likelihood principle, in: Selected Papers of Hirotugu Akaike, Springer, 1998, pp. 199–213.
[38] J.-Y. Zhu, T. Park, P. Isola, A.A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, in: Proceedings of the IEEE International Conference on Computer Vision, pp. 2223–2232.
[39] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. Singh, M.-H. Yang, Diverse image-to-image translation via disentangled representations, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 35–51.
[40] A. Harsh Jha, S. Anand, M. Singh, V. Veeravasarapu, Disentangling factors of variation with cycle-consistent variational auto-encoders, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 805–820.
[41] M. Mirza, S. Osindero, Conditional generative adversarial nets, preprint, arXiv:1411.1784, 2014.
[42] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein GAN, preprint, arXiv:1701.07875, 2017.
[43] L. Yang, D. Zhang, G.E. Karniadakis, Physics-informed generative adversarial networks for stochastic differential equations, preprint, arXiv:1811.02033, 2018.

[44] Y. Pu, L. Chen, S. Dai, W. Wang, C. Li, L. Carin, Symmetric variational autoencoder and connections to adversarial learning, preprint, arXiv:1709.01846, 2017.
[45] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, B. Frey, Adversarial autoencoders, preprint, arXiv:1511.05644, 2015.
[46] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, A. Courville, Adversarially learned inference, preprint, arXiv:1606.00704, 2016.
[47] L. Mescheder, S. Nowozin, A. Geiger, Adversarial variational Bayes: unifying variational autoencoders and generative adversarial networks, preprint, arXiv:1701.04722, 2017.
[48] D.M. Blei, A. Kucukelbir, J.D. McAuliffe, Variational inference: a review for statisticians, J. Am. Stat. Assoc. 112 (2017) 859–877.
[49] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, preprint, arXiv:1412.6980, 2014.
[50] J. Cockayne, C. Oates, T. Sullivan, M. Girolami, Probabilistic meshless methods for partial differential equations and Bayesian inverse problems, preprint, arXiv:1605.07811, 2016.
[51] M. Raissi, P. Perdikaris, G.E. Karniadakis, Inferring solutions of differential equations using noisy multi-fidelity data, J. Comput. Phys. 335 (2017) 736–746.
[52] M. Raissi, P. Perdikaris, G.E. Karniadakis, Numerical Gaussian processes for time-dependent and nonlinear partial differential equations, SIAM J. Sci. Comput. 40 (2018) A172–A198.
[53] J. Cockayne, C. Oates, T. Sullivan, M. Girolami, Bayesian probabilistic numerical methods, preprint, arXiv:1702.03673, 2017.
[54] E. Hopf, The partial differential equation ut+ uux= μxx, Commun. Pure Appl. Math. 3 (1950) 201–230.
[55] D.C. Liu, J. Nocedal, On the limited memory BFGS method for large scale optimization, Math. Program. 45 (1989) 503–528.
[56] A.M. Tartakovsky, C.O. Marrero, D. Tartakovsky, D. Barajas-Solano, Learning parameters and constitutive relationships with physics informed deep neural networks, preprint, arXiv:1808.03398, 2018.
[57] M. White, M. Oostrom, R. Lenhard, Modeling fluid flow and transport in variably saturated porous media with the STOMP simulator, 1: nonvolatile three-phase model description, Adv. Water Resour. 18 (1995) 353–364.
[58] M.T. Van Genuchten, A closed-form equation for predicting the hydraulic conductivity of unsaturated soils, 1: soil, Sci. Soc. Am. J. 44 (1980) 892–898.
[59] I. Tolstikhin, O. Bousquet, S. Gelly, B. Schoelkopf, Wasserstein auto-encoders, preprint, arXiv:1711.01558, 2017.
[60] D.J. Rezende, S. Mohamed, Variational inference with normalizing flows, preprint, arXiv:1505.05770, 2015.
[61] D.P. Kingma, P. Dhariwal, Glow: generative flow with invertible 1 $\times$ 1 convolutions, preprint, arXiv:1807.03039, 2018.
[62] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, 2010, pp. 249–256.