

SPECTRAL METHODS TO STUDY THE ROBUSTNESS OF RESIDUAL NEURAL NETWORKS WITH INFINITE LAYERS

TORSTEN TRIMBORN*, STEPHAN GERSTER AND GIUSEPPE VISCONTI
IGPM, RWTH Aachen University, Germany

ABSTRACT. Recently, neural networks (NN) with an infinite number of layers have been introduced. Especially for these very large NN the training procedure is very expensive. Hence, there is interest to study their robustness with respect to input data to avoid unnecessarily retraining the network.

Typically, model-based statistical inference methods, e.g. Bayesian neural networks, are used to quantify uncertainties. Here, we consider a special class of residual neural networks and we study the case, when the number of layers can be arbitrarily large. Then, kinetic theory allows to interpret the network as a dynamical system, described by a partial differential equation. We study the robustness of the mean-field neural network with respect to perturbations in initial data by applying UQ approaches on the loss functions.

1. Introduction. The use of machine learning algorithms has gained a lot of interest in the past decades [42, 43, 74]. Besides the data science problems like data clustering, regression and classification, image recognition and pattern formation, there are novel applications in the field of engineering as e.g. for production processes [57, 68, 78]. The origins of artificial neural networks date back to the 1940s [33, 53]. Neural networks (NN) have been applied to a huge variety of applications like computer vision, speech recognition or robotics. More recently, also applications to mathematical problems in numerical analysis [62, 63, 83] and optimal control [65] have been studied. The popularity of NN can be partially explained by its data driven ansatz and its simplicity. Having collected sufficient experimental data the NN can be trained and subsequently used as a forward model for nearly all related applications. Most commonly NN are trained by the backpropagation algorithm [30, 46, 84]. Although several studies deal with reducing the computational costs of NN [24, 82], the training procedure is usually computationally expensive especially for large networks. To decide, when a network has to be retrained, quantifying the informative value of the prediction is necessary.

In this work we study the impact of uncertainty in initial data on the NN output. More precisely, we assume that we have a trained NN. We are interested in quantifying the perturbation of the loss function, when the input signal is perturbed, to study the robustness of the network.

2020 *Mathematics Subject Classification.* Primary: 68T07, 68T05; Secondary: 35R60.

Key words and phrases. Neural networks, kinetic equations, uncertainty quantification, stochastic Galerkin, machine learning application.

This research is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC-2023 Internet of Production – 390621612 and supported also by DFG HE5386/18,19 and DFG 320021702/GRK2326. T. Trimborn acknowledge the support by the ERS Prep Fund – Simulation and Data Science.

* Corresponding author: Torsten Trimborn.

This subject has received a lot of attention in the past decade and we refer to [73] for a detailed review. On the one hand, NN have been used to improve UQ techniques [44, 81], but also uncertainty quantification methods have been applied to NN intensively.

Bayesian neural networks provide an extension to posterior inference, which may be seen as maximum likelihood estimation of the weights [79, 3, 29]. Also priors on the weights can be imposed [8]. Approximate Bayesian computation (ABC) forms a class of computational methods, which is linked to Bayesian statistics that can be used to estimate the posterior distributions of model parameters [7, 49, 52, 6, 39].

In all of these model-based statistical inference methods, the likelihood function determines the informative value. Namely, it expresses the probability of the observed data under a particular statistical model.

Typically, the likelihood function causes numerical challenges. ABC methods circumvent this issue by widening the realm of the models for which statistical inference can be considered. However, they are based on assumptions and approximations, which have to be carefully addressed.

In general, these issues become not less challenging if the number of layers increases. For several residual neural networks, however, studying the limit for an infinite number of layers may result in a simplification. Indeed, we obtain for particular residual neural networks in the limit a partial differential equation. Then, uncertainties can be studied by Monte-Carlo, stochastic collocation and stochastic Galerkin methods.

The proposed stochastic Galerkin method describes a priori a functional dependence on the stochastic input. This can be viewed as a prior, where samples may be drawn easily. These are propagated deterministically through the hidden layers of the network. Hence, our approach is closely related to moment matching networks (MMN) [67, 50]. Having trained a MMN, one can also draw easily independent random samples from the output [18, 60]. With our approach, a partial differential equation propagates the uncertainties through the kinetic residual network with infinite layers.

The outline of the paper is as follows. In Section 2 we review the definition of neural networks and especially recall a recently introduced kinetic approach to neural networks. Then, we describe the propagation of uncertainties through the kinetic model. Finally, the UQ methods are presented in Section 3. Especially, we compare the three presented methodologies as possible techniques to quantify the robustness of the residual neural networks. In Section 4, we conduct numerical examples. We present classical machine learning tasks, such as clustering and regression problems to study the robustness of the network with respect to random initial data. In particular, uncertainty quantification is performed on the loss functions. We finish the paper in Section 5 with a brief conclusion and an outlook on future research perspectives.

2. Neural networks and formulation in UQ framework. The crucial building blocks of an artificial neural network are neurons and layers. Each neural network consists at least of one input layer, one hidden layer and one output layer. The input layer $\ell = 0$ is simply characterized by the measurement of d features or input signals $\boldsymbol{x}^0 \in \mathbb{R}^d$. A feature is one type of measured data, e.g. temperature of a tool, length or width of a vehicle, color intensity of an image. The output layer $\ell = L$ contains, after overall application of the network, the output result. Furthermore,

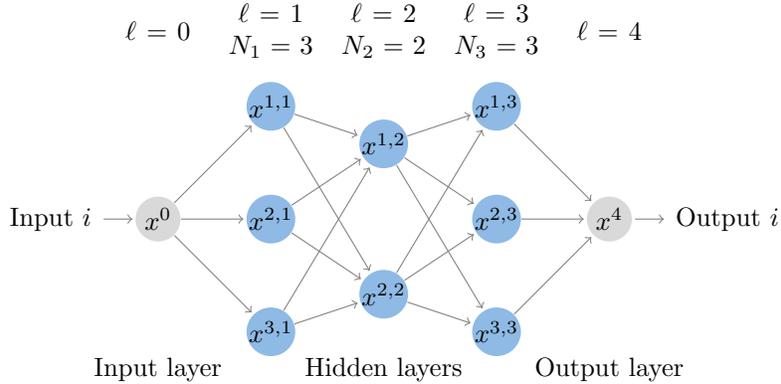


FIGURE 1. Illustration of a neural network with $L + 1 = 5$ total layers.

a neural network may consist of arbitrary many hidden layers $\ell = 1, \dots, L - 1$. The total number of layers, i.e. $L + 1$ including input and output layer, defines the deepness of the network’s structure. The number of neurons in each layer is denoted by N_ℓ . We assume for the input and output layers $N_0 = N_L = 1$. But the number of neurons in each hidden layer $\ell = 1, \dots, L - 1$ may vary. We can imagine each neuron being characterized by a value $\mathbf{x}^{k,\ell} \in \mathbb{R}^d$, $k = 1, \dots, N_\ell$, cf. Figure 1, which is the result of compositions of function blocks defined by an activation function $\sigma : \mathbb{R} \rightarrow \mathbb{R}$. Following [51], the update of a multilayer perceptron neural network can be written in a general fashion as

$$\mathbf{x}^{k,\ell+1} = \sigma \left(\frac{1}{dN_\ell} \sum_{j=1}^{N_\ell} \mathbf{w}_{kj}^\ell \mathbf{x}^{j,\ell} + \mathbf{b}^\ell \right) \quad \text{for } \ell = 1, \dots, L, \quad k = 1, \dots, N_\ell, \quad (1)$$

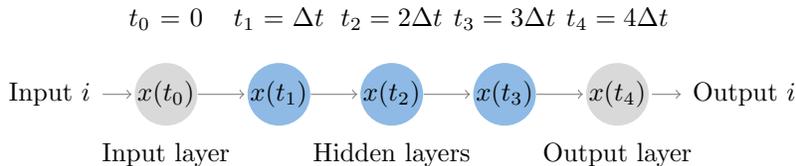
where the weight matrices $\mathbf{w}_{kj}^\ell \in \mathbb{R}^{d \times d}$ and biases $\mathbf{b}^\ell \in \mathbb{R}^d$ are a priori unknown and need to be obtained by training. Here, the activation function is evaluated componentwise. Famous examples of activation functions are the sigmoid $\sigma_S(x) = \frac{1}{1+\exp(-x)}$, ReLu $\sigma_R(x) = \max\{0, x\}$ or hyperbolic tangent $\sigma_T(x) = \tanh(x)$ functions. Weights, biases, as well as the size of a NN, implicitly define the structure of the network. For a comprehensive introduction to NN we refer to [2, 5, 31] and the references therein.

A popular subclass of artificial neural networks are residual neural networks (ResNet), where we are interested in. In comparison to (1) the activation energy of the previous layer is added to the activation function. We assume $N_\ell = N$ for each layer $\ell = 1, \dots, L - 1$. Thus, the update formula of a ResNet [32, 51] is

$$\mathbf{x}^{k,\ell+1} = \mathbf{x}^{k,\ell} + \sigma \left(\frac{1}{dN} \sum_{j=1}^N \mathbf{w}_{kj}^\ell \mathbf{x}^{j,\ell} + \mathbf{b}^\ell \right). \quad (2)$$

So far we have neglected the dependence of the input measurement \mathbf{x}^0 . In case of M different measurements we denote the forward propagation of the i -th measurement by $\mathbf{x}_1^{k,\ell}, \dots, \mathbf{x}_M^{k,\ell}$.

2.1. Training of a network. A crucial part in applying a neural network is the training of the network. Once the structure of the network is established, training

FIGURE 2. Illustration of the SimResNet with $L + 1 = 5$ total layers.

aims to find weights and biases that minimize the distance of the output layer to given targets $\mathbf{h}_i \in \mathbb{R}^d$ with respect to a suitable loss \mathcal{L} , i.e.

$$(\mathbf{W}, \mathbf{B}) \in \operatorname{argmin} \left\{ \frac{1}{M} \sum_{i=1}^M \mathcal{L}(\mathbf{x}_i^L, \mathbf{h}_i) \right\} \quad \text{for } \mathcal{L} : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}_0^+. \quad (3)$$

Here, the collection of weights and biases are denoted by \mathbf{W} and \mathbf{B} . A typical example, see e.g. [38], is the quadratic loss function $\mathcal{L}(\mathbf{x}_i^L, \mathbf{h}_i) = \|\mathbf{x}_i^L - \mathbf{h}_i\|^2$, which is often used for regression problems.

The computational cost of the training procedure depends on the size of the network and may be expensive. Typically, stochastic gradient descent is used to perform the optimization of the parameters [30, 46, 84].

2.2. Kinetic formulation of the ResNet. Recently, studying the limit $L \rightarrow \infty$, i.e. studying deep neural networks with infinite layers, has gained interest in the mathematical community [34, 66, 12]. This allows to represent a discrete network as a continuous function. Similarly to [12] we reformulate the discrete time neural network as a system of continuous ordinary differential equations. This is the intermediate step to formulate the discrete ResNet as a kinetic partial differential equation (PDE) which represents the mathematical cornerstone we consider in this work.

We shortly summarize the derivation of the mean-field equation for a ResNet. The kinetic formulations of a ResNet are derived under the crucial assumption that each hidden layer corresponds to one neuron, i.e. $N_\ell = 1$ for all $\ell = 1, \dots, L - 1$. Then, the number of neurons is fixed and prescribed by the dimension of the input signal only. Therefore, each layer consists of d neurons and each one contains a scalar activation energy, as illustrated in Figure 2.

We introduce the microscopic model SimResNet from [34]. The time evolution of the activation energy of a neuron is for each fixed $i = 1, \dots, M$ defined as

$$\begin{cases} \mathbf{x}_i(t + \Delta t) = \mathbf{x}_i(t) + \Delta t \sigma \left(\frac{1}{d} \mathbf{w}(t) \mathbf{x}_i(t) + \mathbf{b}(t) \right), \\ \mathbf{x}_i(0) = \mathbf{x}_i^0 \end{cases} \quad (4)$$

where the discrete time t identifies the layer. In fact we observe that, compared to (2), the network is reformulated by introducing a parameter $\Delta t > 0$ and a time discrete concept which corresponds to the layer discretization. More precisely, the time step is defined as $\Delta t := \frac{1}{L+1}$. In this way, we can interpret (4) as an explicit Euler discretization of an underlying time continuous model, i.e.

$$\begin{cases} \frac{d}{dt} \mathbf{x}_i(t) = \sigma \left(\frac{1}{d} \mathbf{w}(t) \mathbf{x}_i(t) + \mathbf{b}(t) \right), \\ \mathbf{x}_i(0) = \mathbf{x}_i^0 \end{cases} \quad (5)$$

for each fixed $i = 1, \dots, M$ with $\mathbf{w}(t) \in \mathbb{R}^{d \times d}$, $\mathbf{b}(t) \in \mathbb{R}^d$ being continuous extensions of the discrete optimal weights and biases. The simplifications introduced in the formulation of the SimResNet (4) allow to reduce the complexity of a neural network drastically, and especially the cost of the training stage. A further simplification, considering diagonal matrices $\mathbf{w} \in \mathbb{R}^{d \times d}$, is possible and the resulting network has been shown to provide a suitable approximation of any function $f \in L^1(\mathbb{R}^d)$, see [51].

The special form of the SimResNet is beneficial to compute kinetic formulations. We perform the kinetic limit in the number of measurements M , which means that we consider the case of infinitely many measurements. Since the dimension of a measurement \mathbf{x}_i is directly related to the dimension of the variable of the kinetic distribution function, moderate dimensions d should be considered in applications. According to [34], the mean field equation, corresponding to the dynamics (5), reads

$$\partial_t g(t, \mathbf{x}) + \nabla_{\mathbf{x}} \cdot \left(\sigma \left(\frac{1}{d} \mathbf{w}(t) \mathbf{x} + \mathbf{b}(t) \right) g(t, \mathbf{x}) \right) = 0, \quad (6)$$

where $g : \mathbb{R}_0^+ \times \mathbb{R}^d \rightarrow \mathbb{R}$ is the compactly supported distribution function with normalized initial values

$$g(0, \mathbf{x}) = g_0(\mathbf{x}), \quad \int_{\mathbb{R}^d} g_0(\mathbf{x}) \, d\mathbf{x} = 1.$$

The initial data $g_0(\mathbf{x})$ describe the distribution of the measured data and the mean field equation (6) preserves mass, i.e. $\int_{\mathbb{R}^d} g(t, \mathbf{x}) \, d\mathbf{x} = 1$ for all $t \geq 0$. The derivation is classical and we refer to [27, 37] for general details on mean field theory.

2.3. Propagation of perturbations and robustness of the network in the kinetic limit. We assume a trained NN with a numerically expensive training procedure. Thus, the user of a NN aims to avoid unnecessary training runs. Initial values in real world applications are usually measurements, which are commonly noisy and hence, are perturbations of the true states. We refer to the perturbations as uncertainties and we considered only perturbations of inputs. The NN model propagates the uncertainties forward.

To quantify the forward propagation of the uncertainties through the NN model, and, consequently, the robustness with respect to input signals, we assume a perfectly trained NN, meaning that there have been no measurement errors of the training set nor numerical errors in the training procedure. The M input signals $\mathbf{x}_i^0 \in \mathbb{R}^d$ are perturbed by a d -dimensional random variable $\boldsymbol{\eta}$, which is defined on the probability space $(\Omega, \mathcal{F}(\Omega), \mathbb{P})$. The perturbed input signals with different noise levels $\varepsilon_i > 0$ read as

$$\mathbf{X}_i^0(\omega) = \mathbf{x}_i^0 + \varepsilon_i \boldsymbol{\eta}(\omega) \quad \text{for } i = 1, \dots, M \quad \text{and } \omega \in \Omega.$$

Since we consider a trained NN, the structure of the network does not depend on the uncertainties. Therefore, the dependence of the output signal on the uncertainty can be expressed by $\mathbf{X}_i^L = F(\mathbf{X}_i^0)$, where $F : \mathbb{R}^d \rightarrow \mathbb{R}^d$ denotes the deterministic, pre-trained NN. On the macroscopic level, we look for random solutions

$$g(t, \mathbf{x}, \cdot) : \left(\Omega, \mathcal{F}(\omega) \right) \rightarrow \left(L^1(\mathbb{R}), \mathcal{B}(L^1(\mathbb{R})) \right),$$

where \mathcal{B} denotes the Borel set, that satisfy almost surely (\mathbb{P} -a.s.) the parameterized mean field equation

$$\begin{cases} \partial_t g(t, \mathbf{x}, \boldsymbol{\eta}) + \nabla_{\mathbf{x}} \cdot \left(\sigma \left(\frac{1}{d} \mathbf{w}(t) \mathbf{x} + \mathbf{b}(t) \right) g(t, \mathbf{x}, \boldsymbol{\eta}) \right) = 0, \\ g(0, \mathbf{x}, \boldsymbol{\eta}) = g_0(\mathbf{x}, \varepsilon(\mathbf{x}) \boldsymbol{\eta}). \end{cases}$$

Moreover, we consider solutions that belong for each fixed $g(t, \mathbf{x}, \cdot)$ to the space

$$\mathbb{L}^2(\mathbb{P}) := \left\{ f, g : \Omega \rightarrow \mathbb{R} \mid \langle f, g \rangle_{\mathbb{P}} := \int_{\Omega} f(\omega)g(\omega) \, d\mathbb{P}(\omega), \|f\|_{\mathbb{P}} := \sqrt{\langle f, f \rangle_{\mathbb{P}}} \right\}.$$

3. Uncertainty quantification methods. The research field of uncertainty quantification concerns with the estimation and characterization of uncertainties in the model. Typically, quantifying the impact of uncertain input parameters of a model is of interest. These inputs might be the initial conditions of a PDE or any model parameter.

One may distinguish between intrusive and non-intrusive methods. Non-intrusive methods compute for fixed realisations of the stochastic input the corresponding solutions. Then, the statistics of interest are computed. Since deterministic numerical solvers need not be changed, non-intrusive methods are often preferred in practice. For instance, in the case of partial differential equations finite volume based methods have been proven successful in previous works [69, 76, 17, 80, 54, 77, 64, 1, 55]. Desirable numerical schemes are in smooth regions high-order accurate, but can also resolve singularities in an essentially nonoscillatory (ENO) fashion. CWENO schemes consist of a weighted combination of local reconstructions on different stencils and allow unstructured grids [15, 14, 16, 35, 45, 70].

Numerical cost can be substantially decreased if it is possible to state a priori a functional dependence of the solution on the stochastic input. The dependence can be expressed by orthogonal functions, e.g. orthogonal polynomials, also known as polynomial chaos (gPC) expansions [9, 85, 87].

One large class of UQ methods are described by collocation approaches [72, 86]. They have in common to be non-intrusive and to solve the model exactly for discrete values in the random space. Collocation approaches are very popular and have been applied to a wide range of problems. The simplest method of this class is the well-known and widely used Monte-Carlo method. Another collocation approach is the pseudo spectral or stochastic collocation approach, which can make use of the given functional dependence on the stochastic input.

Stochastic Galerkin methods are intrusive. They compute directly the gPC coefficients that determine the functional dependence. Expansions of the stochastic input are substituted into the governing equations and they are projected to obtain deterministic evolution equations for the gPC coefficients. Applications of this procedure are challenging for general PDEs, since desired properties like hyperbolicity may be lost [59, 19, 21, 22, 47, 58]. The resulting systems for linear hyperbolic [28, 61] and kinetic equations [10, 11, 36, 41, 71, 88, 89] remain well-posed. Furthermore, convergence results of truncated gPC expansions to the true solution can be derived by analyzing the regularity of the solution on the uncertain input [23, 36, 89].

For our applications it is natural to choose collocation approaches in the case of a large artificial NN. The advantage is that any black box solver for the NN can be used. Besides the non-intrusive methods, also the stochastic Galerkin method will be applied to kinetic differential equations.

So far, we have specified the quantity of interest (QoI) by a loss function (3). With the introduction of uncertainties, we obtain the random variables

$$\mathbf{QoI}(\omega) := \int_{\mathbb{R}^d} \mathcal{L}(g(t, \mathbf{x}, \boldsymbol{\eta}(\omega)), h(\mathbf{x})) \, d\mathbf{x} \quad \text{and} \quad \mathbf{QoI}(\omega) := \frac{1}{M} \sum_{i=1}^M \mathcal{L}(\mathbf{X}_i^L(\omega), \mathbf{h}_i).$$

We are interested in statistics such as mean and variance, i.e.

$$\mathbb{E}[\mathbf{QoI}] := \int \mathbf{QoI}(\omega) \, d\mathbb{P}(\omega) \quad \text{and} \quad \mathbb{V}[\mathbf{QoI}] := \mathbb{E}[\mathbf{QoI}^2] - \mathbb{E}[\mathbf{QoI}]^2.$$

In the following we compute these statistics with Monte-Carlo, stochastic collocation and stochastic Galerkin methods. First, we introduce non-intrusive methods, namely Monte-Carlo and stochastic collocation.

3.1. Monte-Carlo methods. For \widehat{M} independent and identically distributed random variables $\mathbf{QoI}_1, \dots, \mathbf{QoI}_{\widehat{M}}$, having the same distribution as the quantity of interest \mathbf{QoI} , we consider the unbiased Monte-Carlo estimators for mean and variance

$$E_{\widehat{M}}[\mathbf{QoI}] := \frac{1}{\widehat{M}} \sum_{i=1}^{\widehat{M}} \mathbf{QoI}_i,$$

$$V_{\widehat{M}}[\mathbf{QoI}] := \frac{1}{\widehat{M} - 1} \sum_{i=1}^{\widehat{M}} \left(\mathbf{QoI}_i - E_{\widehat{M}}[\mathbf{QoI}] \right)^2.$$

Since we are mostly interested in the first and second moments, we consider the root of the mean squared error (RMSE), which satisfies

$$\sqrt{\mathbb{E} \left[\left(E_{\widehat{M}}[\mathbf{QoI}] - \mathbb{E}[\mathbf{QoI}] \right)^2 \right]} = \frac{1}{\sqrt{\widehat{M}}} \mathbb{V}[\mathbf{QoI}].$$

We observe the convergence rate is of order $\mathcal{O}(\widehat{M}^{-1/2})$. This slow convergence is the drawback of the MC method, however, the rate is independent of the dimension of stochastic input. The RMSE can also be decreased by reducing the variance. Typical variance reduction techniques are control variates, importance sampling, antithetic variables [26] and Multilevel-Monte-Carlo methods [25].

3.2. Collocation methods. Under the assumption that the probability measure $d\mathbb{P}(\omega)$ is absolutely continuous with respect to the Lebesgue measure $d\omega$, we can express the moments as the Lebesgue integral

$$\mathbb{E}[\mathbf{QoI}] = \int \mathbf{QoI}(\omega) \, d\mathbb{P}(\omega) = \int f(\mathbf{QoI}(\omega)) \rho(\omega) \, d\omega$$

$$\text{with Radon-Nikodym derivative} \quad \rho(\omega) := \frac{d\mathbb{P}(\omega)}{d\omega},$$

which is almost surely unique defined [56]. Instead of interpreting the quantity of interest \mathbf{QoI} as a random variable, i.e. a measurable function with an associated probability measure, the Radon-Nikodym derivative allows an interpretation as a parameterized function, i.e.

$$\mathbf{QoI} : \Omega_\rho \rightarrow \mathbb{R}^d, \quad \omega \mapsto \mathbf{QoI}(\omega)$$

with associated probability density $\rho(\omega)$. Note that this assumption is rather restrictive, since one cannot define a probability density with respect to an arbitrary measure. For instance, discrete random variables cannot be parameterized in this way.

For many practically relevant probability measures, e.g. normal and uniform distributions, we can compute statistics of interest more efficiently by exploiting

the functional dependence on the stochastic input. More precisely, we will consider projections onto the orthogonal subspaces

$$\hat{\mathcal{S}}_K \subset \mathbb{L}^2(\rho) \quad \text{with} \quad \mathcal{S}_K := \bigoplus_{k=0}^K \hat{\mathcal{S}}_k \longrightarrow \mathbb{L}^2(\rho) \quad \text{for} \quad K \rightarrow \infty.$$

An orthogonal basis of \mathcal{S}_K , denoted as $(\phi_k)_{k=0}^K$, is called generalized polynomial chaos (gPC) basis. Typical choices are

Legendre polynomials with uniform distribution $\rho(\omega) = \frac{1}{2}$, $\Omega_\rho \in [-1, 1]$,
 $\phi_0(\omega) = 1$, $\phi_1(\omega) = \omega$, $\phi_{k+1}(\omega) = \frac{2k+1}{k+1}\omega\phi_k(\omega) - \frac{k}{k+1}\phi_{k-1}(\omega)$,

Hermite polynomials with normal distribution $\rho(\omega) = \frac{1}{\sqrt{2\pi}}e^{-\frac{\omega^2}{2}}$, $\Omega_\rho = \mathbb{R}$,
 $\phi_0(\omega) = 1$, $\phi_1(\omega) = \omega$, $\phi_{k+1}(\omega) = \omega\phi_k(\omega) - k\phi_{k-1}(\omega)$.

Then, the dependency of a quantity of interest on the stochastic input can be expressed as

$$\mathcal{G}_K[\mathbf{QoI}] (\omega) := \sum_{k=0}^K \widehat{\mathbf{QoI}}_k \phi_k(\omega), \quad \widehat{\mathbf{QoI}}_k := \frac{\langle \mathbf{QoI} \phi_k \rangle_\rho}{\|\phi_k\|_\rho^2}. \quad (\text{gPC})$$

With a multi-index $\mathbf{k} := (k_1, \dots, k_d) \in \mathbb{K}$ and an index set $\mathbb{K} \subseteq \mathbb{N}_0^d$ definition (gPC) is extended to the multidimensional case by

$$\mathcal{G}_K[\mathbf{QoI}] (\boldsymbol{\omega}) := \sum_{\mathbf{k} \in \mathbb{K}} \widehat{\mathbf{QoI}}_{\mathbf{k}} \phi_{\mathbf{k}}(\boldsymbol{\omega}) \quad \text{with} \quad \phi_{\mathbf{k}}(\boldsymbol{\omega}) := \phi_{k_1}(\omega_1) \cdots \phi_{k_M}(\omega_d), \quad \boldsymbol{\omega} \in \mathbb{R}^d.$$

Common choices for multidimensional bases, see e.g. [48, 86], are the

$$\begin{aligned} \text{tensor basis } \mathbb{K}_T &:= \{\mathbf{k} \in \mathbb{N}_0^d \mid \|\mathbf{k}\|_0 \leq K\} \quad \text{with} \quad |\mathbb{K}_T| = (K+1)^d, \\ \text{sparse basis } \mathbb{K}_S &:= \{\mathbf{k} \in \mathbb{N}_0^d \mid \|\mathbf{k}\|_1 \leq K\} \quad \text{with} \quad |\mathbb{K}_S| = (d+K)!(d!K!)^{-1}. \end{aligned}$$

In the following, we use the one-dimensional notation (gPC) with $K+1 = |\mathbb{K}|$. According to [9, 13, 20] the truncated expansion converges to the exact solution with respect to the $\mathbb{L}^2(\rho)$ -norm under the mild assumption $\mathbf{QoI} \in \mathbb{L}^2(\rho)$. The rate of convergence, however, depends on the regularity with respect to the stochastic input. In particular for Legendre polynomials, [86, Th. 3.6] gives the estimate

$$\sqrt{\mathbb{E} \left[\left(\mathcal{G}_K[\mathbf{QoI}] - \mathbf{QoI} \right)^2 \right]} \sim \begin{cases} K^{-q} & \text{for } \mathbf{QoI}(\cdot) \in H_\rho^q, \\ e^{-cK} & \text{for } c > 0 \text{ if } \mathbf{QoI}(\cdot) \text{ is analytic.} \end{cases} \quad (7)$$

Here, H_ρ^q denotes the weighted Sobolev space

$$H_\rho^q := \left\{ \mathbf{QoI} \in \mathbb{L}^2(\rho) \mid \sum_{k=0}^q \left\| \partial_\omega^k \mathbf{QoI}(\omega) \right\|_\rho^2 < \infty \right\}.$$

We will exploit this polynomial approximation in several ways.

Gaussian quadrature. A typical approach to compute the integrals in one dimension is Gaussian quadrature with nodes $\xi_1, \dots, \xi_{\widehat{M}}$ and weights $w_1, \dots, w_{\widehat{M}}$. According to [75], we have the error estimates

$$\left| \sum_{i=1}^{\widehat{M}} w_i \mathbf{QoI}(\xi_i) - \mathbb{E}[\mathbf{QoI}(\boldsymbol{\eta})] \right| \sim \begin{cases} \frac{1}{(2\widehat{M})!} & \text{for } \mathbf{QoI}(\cdot) \in C^{2\widehat{M}}, \\ 0 & \text{for } \mathbf{QoI}(\cdot) \in \text{Pol}_{2\widehat{M}-1}, \end{cases}$$

where Pol_p denotes the set of polynomials with degree equal or smaller than $p \in \mathbb{N}_0$. Therefore, this approach can be optimal in one dimension if there is a smooth dependency on the stochastic input.

In multidimensions, however, when the multidimensional functions are represented by full tensors and full grids, the computational complexity grows exponentially with the number of dimensions. Unless the problem structure allows for sparse grids, deterministic quadrature is typically not feasible in higher dimensions.

3.3. Stochastic Galerkin. The stochastic Galerkin (SG) method is built on the idea of the original Galerkin method, i.e. expanding the solution with the help of orthogonal polynomials. In contrast, the underlying equations are projected first. Then, the solution is reconstructed. We consider the general dynamical system $\partial_t \mathbf{QoI}(t, \omega) = \mathcal{D}(\mathbf{QoI}(t, \omega))$. The stochastic Galerkin formulation reads as

$$\partial_t \widehat{\mathbf{QoI}}_k(t) = \frac{1}{\|\phi_k\|_\rho^2} \left\langle \mathcal{D}(\mathcal{G}_K[\mathbf{QoI}](t, \cdot)), \phi_k(\cdot) \right\rangle_\rho \quad \text{for } k = 0, \dots, K.$$

The stochastic Galerkin method consists now in solving these systems of equations. However, there are several challenges. Even if the underlying dynamical system is for each $\omega \in \Omega_\rho$ well-posed, there exist in general no unique solution to the stochastic Galerkin formulation. And even if the stochastic Galerkin formulation is for each fixed $K \in \mathbb{N}_0$ well-posed, the solution does not necessarily converge to the desired solution for $K \rightarrow \infty$. We can resolve these issues for solutions to general, but *linear* kinetic initial value problems

$$\partial_t g(t, \mathbf{x}, \omega) + \mathcal{T} \nabla_{\mathbf{x}} \cdot g(t, \mathbf{x}, \omega) = \mathcal{Q}g(t, \mathbf{x}, \omega) \quad \text{with } g(0, \mathbf{x}, \omega) = g_0(\mathbf{x}, \omega), \quad (8)$$

where the uncertainty arises only from initial values. This means for the mean field equation $\mathcal{T} = \sigma(\frac{1}{d} \mathbf{w}(t) \mathbf{x} + \mathbf{b}(t))$, $\mathcal{Q} = \frac{1}{d} \mathbf{w}(t) \sigma'(\frac{1}{d} \mathbf{w}(t) \mathbf{x} + \mathbf{b}(t))$. Then, the stochastic Galerkin formulation consists of $K + 1$ decoupled systems

$$\partial_t \hat{g}_k(t, \mathbf{x}) + \mathcal{T} \nabla_{\mathbf{x}} \cdot \hat{g}_k(t, \mathbf{x}) = \mathcal{Q} \hat{g}_k(t, \mathbf{x}) \quad \text{with } \hat{g}_k(0, \mathbf{x}) := \frac{\langle g_0(\mathbf{x}, \cdot) \phi_k(\cdot) \rangle_\rho}{\|\phi_k\|_\rho^2} \quad (9)$$

and is hence well-posed. Moreover, the convergence for $K \rightarrow \infty$ should be relatively fast. We consider L^2 -solutions [4, Def. A.3] to the linear, kinetic equations (8) and (9). These are formally given by

$$g(t, \mathbf{x}, \omega) = \mathcal{M}(t)g_0(\mathbf{x}, \omega) \quad \text{and} \quad \hat{g}_k(t, \mathbf{x}) = \mathcal{M}(t)\hat{g}_k(0, \mathbf{x}).$$

If the solution operator $\mathcal{M}(t)$ is a continuous and hence bounded operator, satisfying $\|\mathcal{M}(t)\| < \infty$ for all $t \geq 0$, we have the a priori estimate

$$\begin{aligned} \sqrt{\mathbb{E} \left[\|\mathcal{G}_K[g](t, \cdot, \boldsymbol{\eta}) - g(t, \cdot, \boldsymbol{\eta})\|^2 \right]} &= \|\mathcal{M}(t)\| \sqrt{\sum_{k=K+1}^{\infty} \|\hat{g}_k(0, \cdot)\|^2} \\ &\sim \begin{cases} K^{-p} & \text{for } g_0(x, \cdot) \in H_\rho^q, \\ e^{-cK} & \text{for } c > 0 \text{ if } g_0(x, \cdot) \text{ is analytic.} \end{cases} \end{aligned}$$

Hence, for linear kinetic equations, this approach can provide spectral convergence if initial values depend sufficiently smooth on the random input.

4. Numerical examples. In this section we use the previous methods to study the behavior of uncertainty in the initial input of a neural network model to quantify its robustness by measuring the deviation from the correct target as function of the size of the uncertainty.

In particular, we apply Monte-Carlo and stochastic collocation to the discrete deep artificial neural network, and we apply stochastic Galerkin to the PDE formulation of the deep residual neural network.

We focus on regression and classification problems, which are tasks belonging to the field of supervised machine learning. Both are based on using known datasets, usually named as training datasets, to make predictions. The main difference between the two types of problems is that the output variable in regression is numerical or continuous, while the output for classification is categorical or discrete.

4.1. Deep artificial neural network.

4.1.1. Regression problem. In machine learning, a regression task aims to estimate or approximate a function $h : \mathbb{H} \subseteq \mathbb{R}^d \rightarrow \mathbb{R}^d$, mapping the input variables $\mathbf{x} \in \mathbb{H}$ to the numerical or continuous output variables $\mathbf{y} = h(\mathbf{x}) \in \mathbb{R}^d$. In the following we consider one dimension $d = 1$ and $h(x) = \tanh(x)$ on $\mathbb{H} = [-4, 4]$.

We make use of the discrete model (1) with 4 total layers and 2 hidden layers, each one having $N_1 = N_2 = 4$ neurons. Moreover, the activation function is chosen as $\sigma(x) = \sigma_T(x) = \tanh(x)$. The training dataset is artificially built using $M = 50$ equally spaced points $x_i^0 \in \mathbb{H}$, $i = 1, \dots, M$ as input signals with corresponding targets $y_i = h(x_i^0)$. The network is trained using the gradient descent method as backpropagation algorithm with learning rate 0.1. The training procedure determines the optimal weights and bias of the network. The result of the training step is observed in the top left panel of Figure 3, where the blue circles represent the true targets y_i , $i = 1, \dots, M$, while the red stars are the estimated values by the trained network.

Once we have trained the network, we store the optimal parameters to perform the forward propagation of the network with given noisy input signals. The uncertain perturbation is added to the input data of the training dataset, as described in Section 2.3. We consider a uniformly distributed random variable on $[-1, 1]$, i.e. $\boldsymbol{\eta} \sim \mathcal{U}(-1, 1)$. We recall that the strength of the perturbation is modeled by the multiplicative factor $\varepsilon > 0$.

We apply Monte-Carlo and stochastic collocation to compute the evolution of the network under the uncertain inputs. In particular, with Monte-Carlo we consider $\widehat{M} = 1000$ random samples from the given uniform distribution of the uncertain parameter. Instead, for stochastic collocation we perturb the true input data along the 4 collocation points of the Legendre polynomials. In both cases, we consider 10 equally spaced values of the noise level ε , from 0.1 to 1. We compute the expected value and the variance of the quantity of interest, which gives information on the L^2 -distance between the true target and the output of the network under uncertain input data.

In the top left panel of Figure 3 we show the expected value and the standard deviation from the expected value of the pointwise error, namely computed on each data input. In Figure 3 we provide the expected value and the variance as functions of the noise level ε , in the top right panel and in the bottom left panel respectively. As we expect, the error decreases with the noise level. Moreover, the bottom right panel shows the convergence rate of the Monte-Carlo method as a function of the

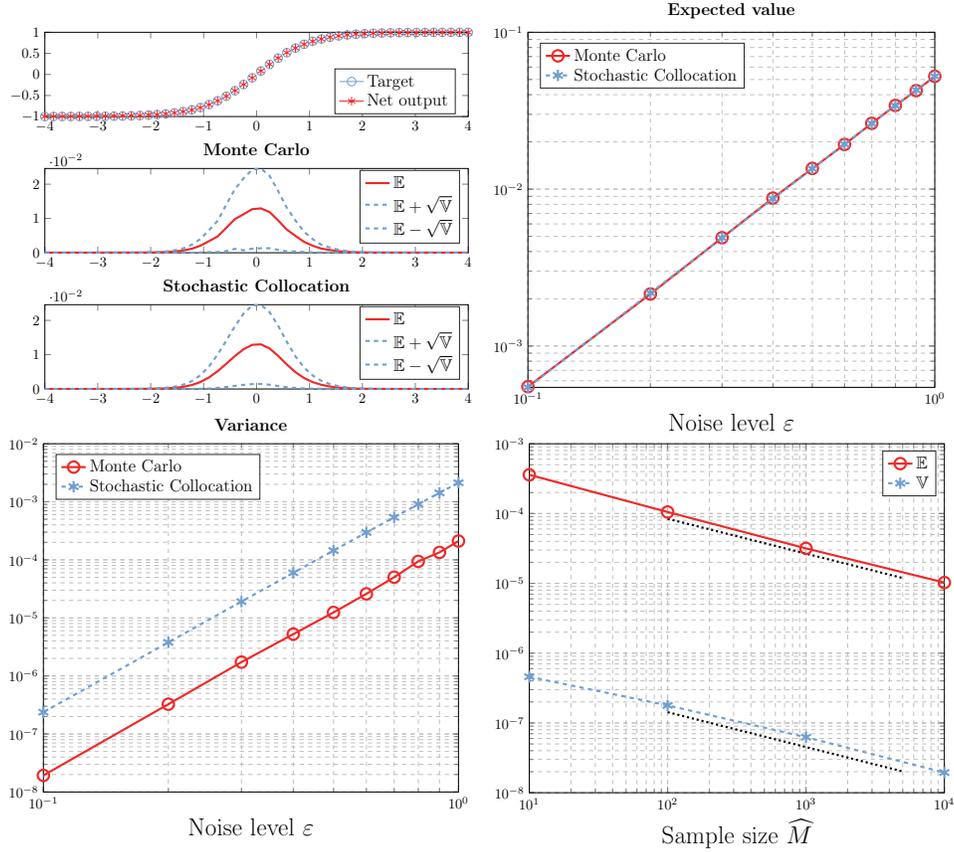


FIGURE 3. Top left panel: Regression of the target function $h(x) = \tanh(x)$ on $\mathbb{H} = [-4, 4]$ provided by the deep artificial neural networks with the statistical quantities computed with the Monte-Carlo method and the stochastic collocation method with noise level $\varepsilon = 0.3$. Top right panel: Expected value of the quantity of interest. Bottom left panel: Variance of the quantity of interest. Bottom right panel: Convergence rate of the Monte-Carlo method.

number of samples \widehat{M} with fixed value of the noise level $\varepsilon = 0.3$. The black dotted lines represent the rate $1/2$.

4.1.2. *Classification problem.* On the other hand, classification problems attempt to estimate a function $h : \mathbb{H} \subseteq \mathbb{R}^d \rightarrow \mathcal{C}$ from the input variables $\boldsymbol{x} \in \mathbb{H}$ of the dataset to discrete or categorical target variables $y = h(\boldsymbol{x}) \in \mathcal{C}$. Here \mathcal{C} is a discrete set of the categorical variables. In the following we consider $d = 1$ and

$$h(x) = \begin{cases} 0, & 3 \leq x \leq 5, \\ 1, & 5 < x \leq 8. \end{cases}$$

From an application point of view, this experiment can be seen as the identification problem of the type of a vehicle, such as car or truck, depending on the

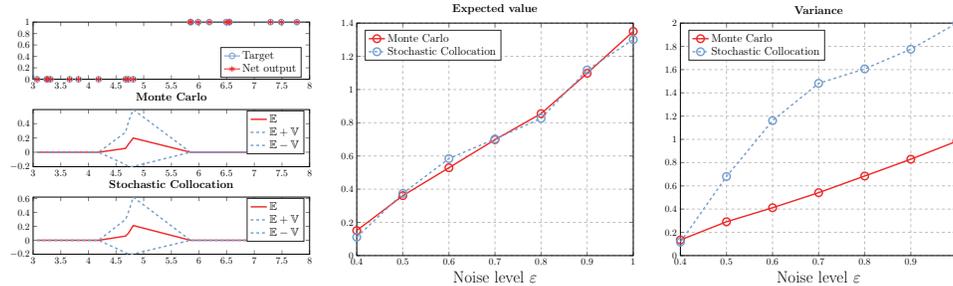


FIGURE 4. Top left panel: classification problem with the statistical quantities computed with the Monte-Carlo method and the stochastic collocation method with noise level $\varepsilon = 0.5$. Top right panel: expected value of the quantity of interest. Bottom left panel: variance of the quantity of interest. Bottom right panel: convergence rate of the Monte-Carlo method.

measurement of its length. Here, we consider cars having label 0 and trucks having label 1. The interesting question in applications is how low should be the measurement noise level in order to have, for instance, an error below to 5% of wrong classifications.

We make use of the discrete model (1), with $L = 3$, which means 4 total layers and 2 hidden layers, each one having $N_1 = N_2 = 4$ neurons. Moreover, the activation function is chosen as $\sigma(x) = \sigma_T(x) = \tanh(x)$.

The training dataset is artificially built using $M = 20$ uniformly distributed points in $\mathbb{H} = [3, 8]$ representing the input signals x_i^0 with targets $y_i = h(x_i^0)$ for $i = 1, \dots, M$. We again train the network using the gradient descent method with learning rate 0.1. The result of the training step is observed in the left panel of Figure 4, where the blue circles represent the true targets y_i , $i = 1, \dots, M$, while the red stars are the estimated values by the trained network.

The uncertain perturbation is added by considering a uniformly distributed random variable on $\boldsymbol{\eta} \sim \mathcal{U}(-1, 1)$. For the strength of the perturbation, modeled by the multiplicative factor ε , we choose 7 equally spaced values between 0.4 and 1.

We perform the evolution of the network under the uncertain inputs using Monte-Carlo and stochastic collocation. In particular, with Monte-Carlo we consider $\widehat{M} = 1000$ random samples from the given uniform distribution of the uncertain parameter. Instead, for stochastic collocation we perturb the true input data along the 4 collocation points of the Legendre polynomials. We compute the expected value and the variance of the quantity of interest that gives information on the averaged total number of wrong classifications.

In the left panel of Figure 4 we show the expected value and the standard deviation from the expected value of the pointwise error, namely computed on each data input. In Figure 4 we provide also the plot of the expected value and of the variance as function of the noise level ε , in the center panel and in the right panel respectively. As we expect, the error decreases with the noise level. We observe that, for instance, to have less than 5% of wrong classifications, the noise level needs to be less than 0.6.

4.2. **Kinetic ResNet.** Regression problems based on the random mean field equation (6) consist of the following steps.

- 1) Determine gPC modes for initial data by an orthogonal projection.
- 2) Solve the stochastic Galerkin formulation (9).
- 3) Reconstruct the statistics of interest.

We will perform these steps for the following test cases, where a uniform distributed random variable $\boldsymbol{\eta} \sim \mathcal{U}(-1, 1)$ and normalized Legendre polynomials are used. We consider

$$\begin{aligned} \text{CASE 1:} \quad & \partial_t g(t, x, \boldsymbol{\eta}) + \partial_x \left(\tanh(x - 5)g(t, x, \boldsymbol{\eta}) \right) = 0 \quad \text{for } x \in (2, 8), \\ & g_0(x, \boldsymbol{\eta}) = \begin{cases} \frac{1}{4}, & x \in [3 + \boldsymbol{\eta}, 7 + \boldsymbol{\eta}], \\ 0, & \text{else,} \end{cases} \end{aligned}$$

where zero flux boundary conditions are used, and

$$\begin{aligned} \text{CASE 2:} \quad & \partial_t g(t, x, \boldsymbol{\eta}) + \partial_x \left(\tanh(x - 5)g(t, x, \boldsymbol{\eta}) \right) = 0 \quad \text{for } x \in (-4, 6), \\ & g(t, -4, \boldsymbol{\eta}) = g(t, 6, \boldsymbol{\eta}), \\ & g_0(x, \boldsymbol{\eta}) = \frac{1}{\sqrt{2\pi}(1 + \boldsymbol{\eta}/2)} \exp\left(-\frac{1}{2}\left(\frac{x - 1}{1 + \boldsymbol{\eta}/2}\right)^2\right). \end{aligned}$$

Orthogonal projection of initial values. In CASE 2, there is a smooth dependence on the stochastic input, i.e. $g_0(x, \cdot) \in C^\infty$, whereas we have only $g_0(x, \cdot) \in L^2$ in CASE 1. Therefore, the gPC modes $\hat{g}_k(0, x) := \langle g_0(x, \cdot) \phi_k(\cdot) \rangle_\rho$ are computed using a standard Monte-Carlo method with 10^5 samples in CASE 1 and with Gaussian quadrature with 100 nodes in CASE 2.

The upper panel of Figure 5 shows exact realisations of initial data for CASE 1 in black being in the confidence region (gray), which contains all realisations. The mean (red) and the variance (blue) are calculated exactly by $\mathbb{V}[g_0(x, \boldsymbol{\eta})] = \mathbb{E}[g_0^2(x, \boldsymbol{\eta})] - \mathbb{E}[g_0(x, \boldsymbol{\eta})]^2$ and

$$\mathbb{E}[g_0^m(x, \boldsymbol{\eta})] = \frac{1}{4^m} \mathbb{P}\left[3 + \frac{\boldsymbol{\eta}}{2} \leq x \leq 5 + \frac{\boldsymbol{\eta}}{2}\right] = \frac{1}{4^m} \left[1 - F_\rho(2x - 10) - F_\rho(2x - 6)\right]$$

with distribution function $F_\rho(x) = \begin{cases} 0, & x \leq 2, \\ \frac{1}{4}(x - 2), & 2 < x \leq 5, \\ 1, & 5 \leq x. \end{cases}$

We observe that the moments are smooth although realisations are discontinuous. This allows a relatively accurate approximation by superpositions of smooth gPC modes (green), whose evolution is described by equation (9). Indeed, the exact variance is well approximated from below by $\|\hat{g}_1(0, x), \dots, \hat{g}_K(0, x)\|^2 \nearrow \mathbb{V}[g_0(x, \boldsymbol{\eta})]$ for $K \rightarrow \infty$, as illustrated in blue with respect to the right y -axis. The second and third panel show the root of the mean squared error (RMSE). Due to the non-smooth dependence on the stochastic input we observe the slow decay rate $1/2$, which is similar to a Monte-Carlo method. Figure 6 is devoted to CASE 2. Here the RMSE decays faster. We even observe exponential decay as guaranteed by the estimate (7). As reference we have used the truncation $K = 15$.

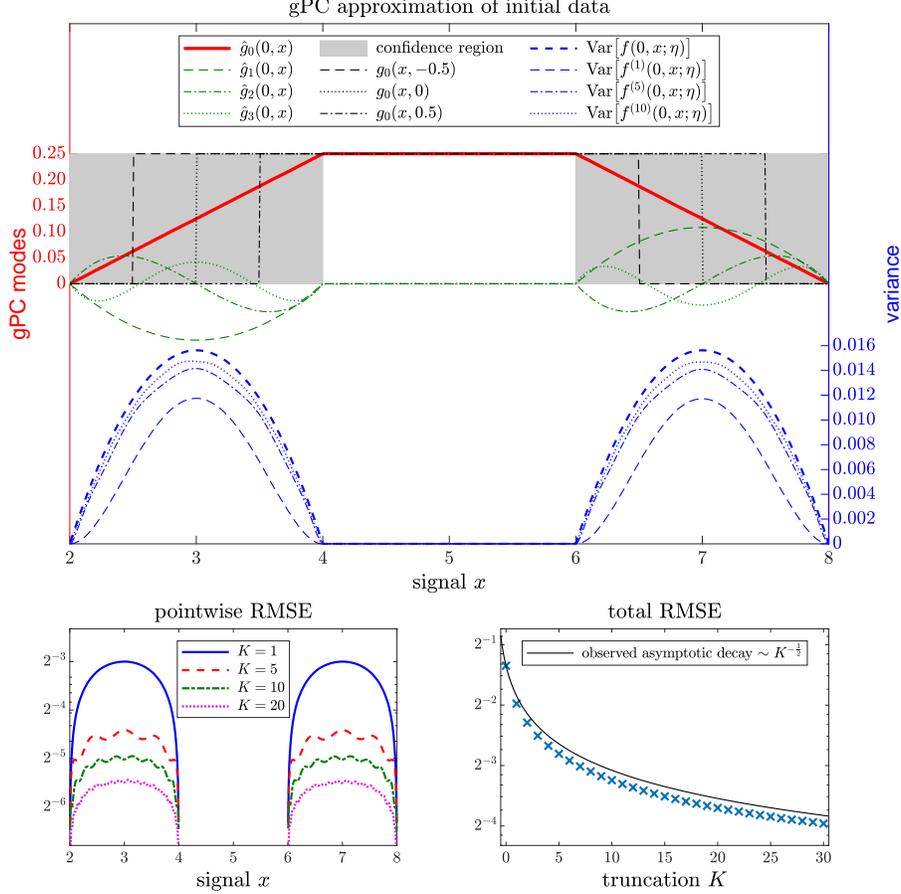


FIGURE 5. CASE 1: Projection of initial data and root mean squared error (RMSE) with exactly computed mean and variance. Upper panel: The left y -axis shows the gPC modes $\hat{g}_k(0, x) := \langle g_0(x, \cdot) \phi_k(\cdot) \rangle_\rho$ of initial values. The mean $\mathbb{E}[g_0(x, \boldsymbol{\eta})] = \hat{g}_0(0, x)$ is plotted in red, the others in green. The confidence region, which contains \mathbb{P} -a.s. all realisations $g_0(x, \boldsymbol{\eta}(\omega))$ is gray shaded. The right y -axis shows the variance corresponding to different gPC truncations. The upper left panel shows for each fixed signal x the root mean squared error (RMSE) of the gPC truncations. The right panel shows the total RMSE for various gPC truncations.

Solving the stochastic Galerkin formulation. To approximate the PDEs (9) a discretization $\Delta x > 0$ is used to divide an interval $\mathbb{H} = [x_{\min}, x_{\max}]$ into N cells $C_j := [x_{\min} + (j-1)\Delta x, x_{\min} + j\Delta x]$, $j = 1, \dots, N$ such that $\mathbb{H} = C_1 \cup \dots \cup C_N$. The evolution of cell averages is simulated by a third-order CWENO-reconstruction from [15]. It uses a local Lax-Friedrichs flux and a strong stability preserving (SSP) Runge-Kutta method with three stages [40].

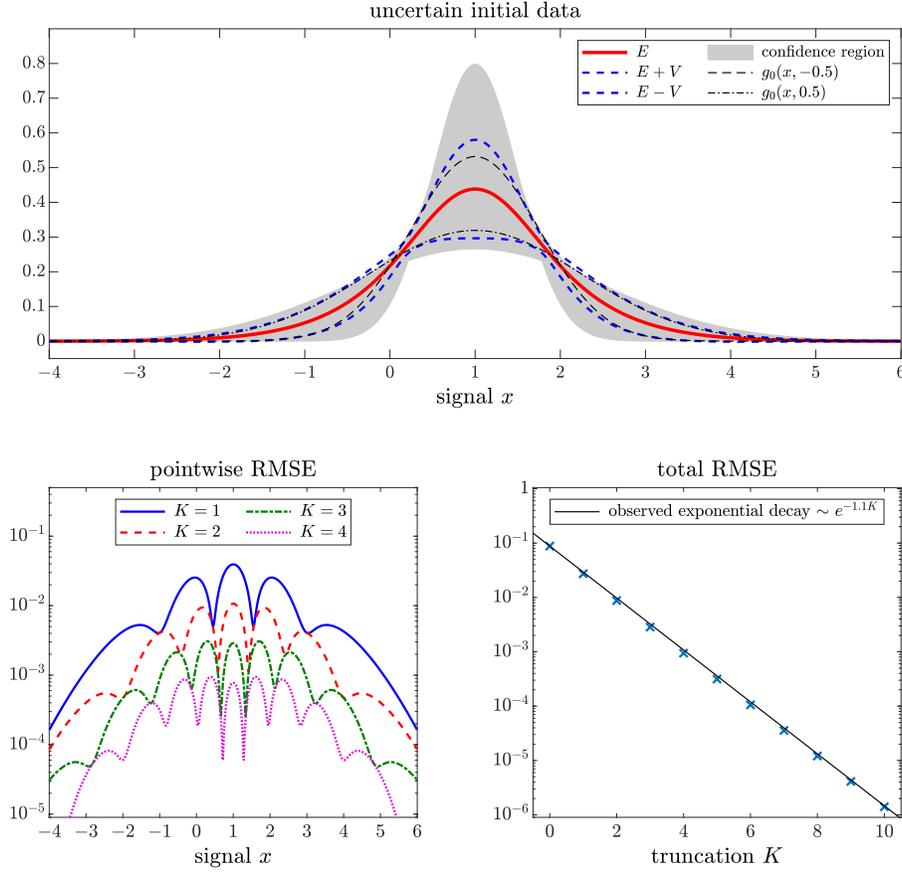


FIGURE 6. CASE 2: Projection of initial data and root mean squared error (RMSE) with exactly computed mean and variance. Upper panel: The blue lines describe deviations from the mean of initial data (red). Two realisations are shown in black, which are in the confidence region (gray shaded). The lower left panel shows for each fixed signal x the root mean squared error (RMSE) of the gPC truncations. The right panel shows the total RMSE for various gPC truncations.

The upper panels of Figure 7 show that the solution to CASE 1 converges to a delta distribution in $x \in \{2, 8\}$. The lower panels investigate the probability distribution for a fixed signal. The distribution is determined by a `Matlab` built-in kernel density estimator with 10^5 samples from the truncated gPC expansion. Figure 8 shows the time evolution of the gPC modes for CASE 2, likewise. Here the solution converges to a delta distribution in $x = 1$.

Reconstructing the statistics of interest. The target of CASE 1 is characterized by two delta distributions located at the binary values $x \in \{2, 8\}$, the target of CASE 2

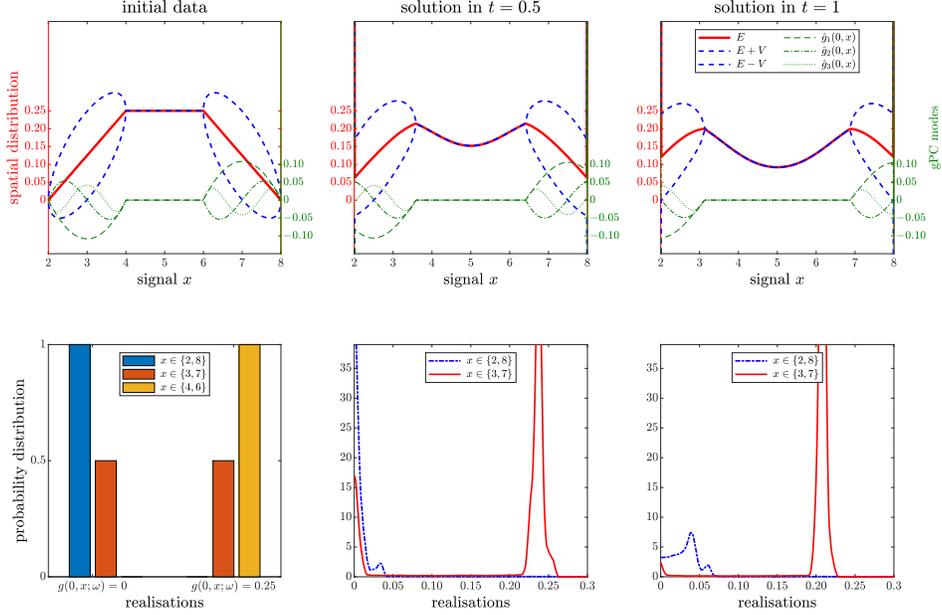


FIGURE 7. CASE 1: Solutions to the stochastic Galerkin formulation with truncation $K = 30$ and discretization $\Delta x = 0.01$. The upper panels show in red the first gPC mode $\hat{g}_0(t, x)$, which describes the temporal evolution of the mean. The other modes are plotted in green, which determine the spread of the stochastic perturbations. The lower panels show the probability distributions of the random quantity $g(t, x, \eta)$ for a fixed signal $x \in \{2, 8\}$ at time $t = 0$ (left), $t = 0.5$ (middle) and $t = 1$ (right).

is a delta distribution at $x = 1$. If N is even, the discrete analogues read as

$$h(x) = \begin{cases} \frac{1}{2\Delta x}, & x \in C_1 \cup C_N, \\ 0, & \text{else} \end{cases} \quad \text{in CASE 1,}$$

$$h(x) = \begin{cases} \frac{1}{2\Delta x}, & x \in C_{N/2} \cup C_{N/2+1}, \\ 0, & \text{else} \end{cases} \quad \text{in CASE 2.}$$

We are interested in the random loss

$$\mathbf{QoI}(t, x, \eta) = \left(\mathcal{G}_K[g](t, x, \eta) - h(x) \right)^2.$$

By using the multinomial theorem, we can express moments exactly as

$$\mathbb{E} \left[\mathbf{QoI}(t, x, \eta)^{2n} \right] = \sum_{\|\mathbf{k}\|_1=2n} \frac{C_{\mathbf{k}}(2n)!}{k_0! \cdots k_K!} \left(\hat{g}_0(t, x) - h(x) \right)^{k_0} \prod_{\ell=1}^K \hat{g}_\ell^{k_\ell}(t, x)$$

$$\text{with precomputed constants } C_{\mathbf{k}} := \mathbb{E} \left[\phi_0^{k_0}(\eta) \cdots \phi_K^{k_K}(\eta) \right].$$

In particular, we have $\mathbb{E}[\mathbf{QoI}(\eta)] = \|\hat{g}_0 - h, \hat{g}_1, \dots, \hat{g}_K\|^2$. The gPC expansion $\mathcal{G}_K[g](t, x, \varepsilon\eta)$ gives also samples for general loss functions and noise levels.

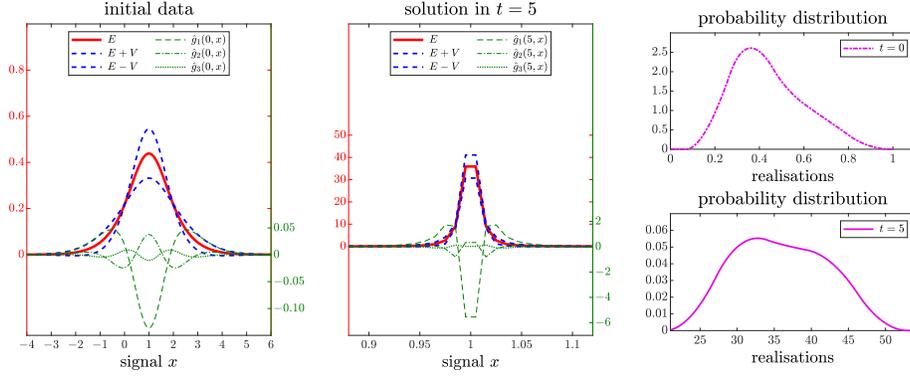


FIGURE 8. CASE 2: Solutions to the stochastic Galerkin formulation with truncation $K = 10$ and discretization $\Delta x = 0.01$. The left y -axes of the first and second panel show the mean and deviations. The right y -axes show the gPC modes $\hat{g}_1, \dots, \hat{g}_3$. The right panels show the probability distributions of the random quantity $g(t, 1, \boldsymbol{\eta})$ at time $t = 0$ (left), $t = 0$ and $t = 5$.

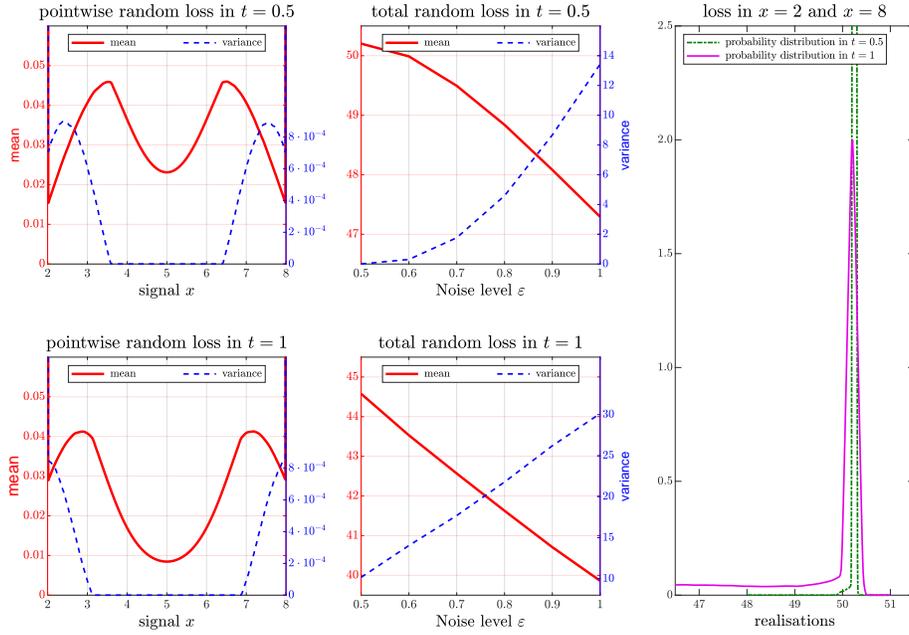


FIGURE 9. CASE 1: Mean, variance and distribution for random loss. The left panels show for each fixed signal x the mean and the variance of the random loss. The middle panels show for various noise levels ε the mean and variance of the total random loss. The right panel shows the probability distributions of the random loss in $x = 2$ and $x = 8$ at time $t = 0.5$ and $t = 1$.

Figure 9 and 10 show the mean, variance and the probability distribution for a fixed signal $x \in \mathbb{H}$, where the target is a delta distribution. In CASE 1, a higher noise level lets realisations be closer to the target. Hence, the mean of quadratic deviations increases for lower noise levels. In contrast, the mean decreases in CASE 2 with the noise level, since realisations are closer to the target for small noise levels.

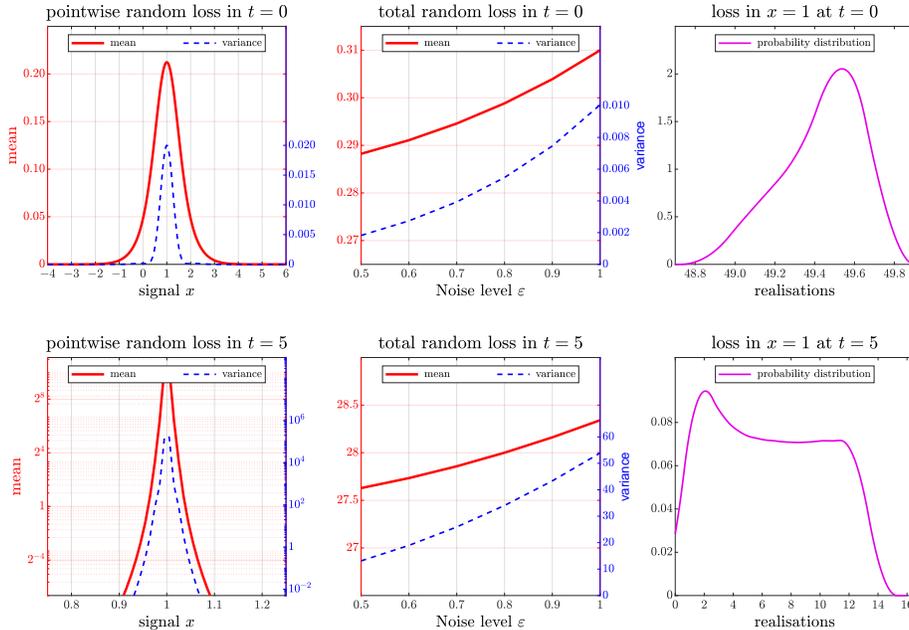


FIGURE 10. CASE 2: Mean, variance and distribution for random loss. The left panels show for each fixed signal x the mean and the variance of the random loss. The middle panels show for various noise levels ε the mean and variance of the total random loss. The right panels show the probability distributions of the random loss in $x = 1$, respectively at time $t = 0$ and $t = 5$.

5. Conclusion. In this work, we have proposed a technique to use UQ approaches to quantify the robustness of the model output of a neural network due to the influence of uncertain inputs. The robustness is measured by the expected distance and variance of the output to the true target. In particular, the cornerstone of this work is the use of the mean-field formulation of a discrete time residual neural network. Several numerical tests have been applied by employing Monte-Carlo, Stochastic Collocation and Stochastic Galerkin methods. Here, uncertainties arise only from input data. Uncertainties in the model structure and predictive uncertainty are topics of interest as well, and possibly subjects of future research.

Acknowledgments. This research is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC-2023 Internet of Production – 390621612 and by DFG HE5386/18,19, DFG 320021702/GRK2326. Torsten Trimborn acknowledges the support by the ERS Prep Fund – Simulation and Data Science.

REFERENCES

- [1] R. Abgrall and S. Mishra, Uncertainty quantification for hyperbolic systems of conservation laws, Handbook of numerical methods for hyperbolic problems, *Handb. Numer. Anal.*, Elsevier/North-Holland, Amsterdam, **18** (2017), 507–544.
- [2] D. Anderson and G. McNeill, Artificial neural networks technology, *Kaman Sciences Corporation*, **258** (1992), 1–83.
- [3] T. Auld, A. W. Moore and S. F. Gull, Bayesian neural networks for internet traffic classification, *IEEE Transactions on Neural Networks*, **18** (2007), 223–239.
- [4] G. Bastin and J.-M. Coron, *Stability and Boundary Stabilization of 1-d Hyperbolic Systems*, 1st edition, Progress in nonlinear differential equations and their applications, Birkhäuser, Switzerland, 2016.
- [5] H. D. Beale, H. B. Demuth and M. Hagan, Neural network design, *PWS*, Boston.
- [6] M. A. Beaumont, [Approximate bayesian computation](#), *Annu. Rev. Stat. Appl.*, **6** (2019), 379–403.
- [7] M. G. B. Blum, [Approximate bayesian computation: A nonparametric perspective](#), *J. Amer. Statist. Assoc.*, **105** (2010), 1178–1187.
- [8] C. Blundell, J. J. Cornebise, K. Kavukcuoglu and D. Wierstra, Weight uncertainty in neural networks, in *ICML*, 2015.
- [9] R. H. Cameron and W. T. Martin, [The orthogonal development of non-linear functionals in series of Fourier-Hermite functionals](#), *Ann. of Math.*, **48** (1947), 385–392.
- [10] J. A. Carrillo, L. Pareschi and M. Zanella, [Particle based gPC methods for mean-field models of swarming with uncertainty](#), *Commun. Comput. Phys.*, **25** (2019), 508–531.
- [11] J. A. Carrillo and M. Zanella, [Monte Carlo gPC methods for diffusive kinetic flocking models with uncertainties](#), *Vietnam J. Math.*, **47** (2019), 931–954.
- [12] R. T. Q. Chen, Y. Rubanova and J. B. D. K. D. Duvenaud, Neural ordinary differential equations, in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., **31** (2018), 6571–6583.
- [13] R. Courant and D. Hilbert, *Methods of Mathematical Physics*, Vol. II. Partial differential equations. Reprint of the 1962 original. Wiley Classics Library. A Wiley-Interscience Publication. John Wiley & Sons, Inc., New York, 1989.
- [14] I. Cravero, G. Puppo, M. Semplice and G. Visconti, [Cool WENO schemes](#), *Comput. & Fluids*, **169** (2018), 71–86.
- [15] I. Cravero, G. Puppo, M. Semplice and G. Visconti, [CWENO: Uniformly accurate reconstructions for balance laws](#), *Math. Comp.*, **87** (2018), 1689–1719.
- [16] I. Cravero, M. Semplice and G. Visconti, [Optimal definition of the nonlinear weights in multidimensional central WENOZ reconstructions](#), *SIAM J. Numer. Anal.*, **57** (2019), 2328–2358.
- [17] R. Crisovan, D. Torlo, R. Abgrall and S. Tokareva, [Model order reduction for parametrized nonlinear hyperbolic problems as an application to uncertainty quantification](#), *J. Comput. Appl. Math.*, **348** (2019), 466–489.
- [18] Y. Deng, Y. Shen, K. Chen and H. Jin, Training recurrent neural network through moment matching for nlp applications, (2018), 3353–3357.
- [19] B. Després, G. Poëtte and D. Lucor, [Robust uncertainty propagation in systems of conservation laws with the entropy closure method](#), *Uncertainty Quantification in Computational Fluid Dynamics*, Lect. Notes Comput. Sci. Eng., Springer, Heidelberg, **92** (2013), 105–149.
- [20] D. Funaro, *Polynomial Approximation of Differential Equations*, Lecture Notes in Physics. New Series m: Monographs, 8. Springer-Verlag, Berlin, 1992.
- [21] S. Gerster and M. Herty, Entropies and symmetrization of hyperbolic stochastic Galerkin formulations, *Communications in Computational Physics*, **27** (2020), 639–671.
- [22] S. Gerster, M. Herty and A. Sikstel, [Hyperbolic stochastic Galerkin formulation for the p-system](#), *J. Comput. Phys.*, **395** (2019), 186–204.
- [23] S. Gerster, M. Herty and H. Yu, Hypocoercivity of stochastic Galerkin formulations for stabilization of kinetic equations, *RWTH*, preprint, 1–21.
- [24] F. S. Gharehchopogh, [Neural networks application in software cost estimation: A case study](#), in *2011 International Symposium on Innovations in Intelligent Systems and Applications*, IEEE, (2011), 69–73.
- [25] M. B. Giles, [Multilevel Monte Carlo methods](#), *Springer Proceedings in Mathematics & Statistics*, Monte Carlo and Quasi-Monte Carlo Methods, **65** (2012), 83–103.

- [26] P. Glasserman, *Monte Carlo Methods in Financial Engineering*, Springer, New York, 2004.
- [27] F. Golse, [On the dynamics of large particle systems in the mean field limit](#), in *Macroscopic and Large Scale Phenomena: Coarse Graining, Mean Field Limits and Ergodicity*, Lect. Notes Appl. Math. Mech., Springer, **3** (2016), 1–144.
- [28] D. Gottlieb and D. Xiu, Galerkin method for wave equations with uncertain coefficients, *Commun. Comput. Phys.*, **3** (2008), 505–518.
- [29] A. Graves, Practical variational inference for neural networks, *NIPS*, (2011), 2348–2356.
- [30] E. Haber, F. Lucka and L. Ruthotto, Never look back - A modified EnKF method and its application to the training of neural networks without back propagation, 2018, Preprint, [arXiv:1805.08034](#).
- [31] M. H. Hassoun et al., *Fundamentals of Artificial Neural Networks*, MIT press, 1995.
- [32] K. He, X. Zhang, S. Ren and J. Sun, [Deep residual learning for image recognition](#), *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.
- [33] D. O. Hebb, *The Organization of Behavior: A Neuropsychological Theory*, Psychology Press, 2005.
- [34] M. Herty, T. Trimborn and G. Visconti, Kinetic theory for residual neural networks, arXiv preprint, [arXiv:2001.04294](#).
- [35] G. Hu, R. Li and T. Tang, [A robust WENO type finite volume solver for steady Euler equations on unstructured grids](#), *Commun. Comput. Phys.*, **9** (2011), 627–648.
- [36] J. Hu and S. Jin, [A stochastic Galerkin method for the Boltzmann equation with uncertainty](#), *J. Comput. Phys.*, **315** (2016), 150–168.
- [37] P.-E. Jabin, [A review of the mean field limits for vlasov equations](#), *Kinetic & Related Models*, **7** (2014), 661–711.
- [38] K. Janocha and W. M. Czarnecki, On loss functions for deep neural networks in classifications, 2017, Preprint, [arXiv:1702.05659v1](#).
- [39] B. Jiang, T.-Y. H Wu, C. Zheng and W. Wong, Learning summary statistic for approximate bayesian computation via deep neural network, *Statistica Sinica*, **27** (2017), 1595–1618.
- [40] G.-S. Jiang and C.-W. Shu, [Efficient implementation of weighted ENO schemes](#), *J. Comput. Phys.*, **126** (1996), 202–228.
- [41] S. Jin and Y. Zhu, [Hypocoercivity and uniform regularity for the Vlasov-Poisson-Fokker-Planck system with uncertainty and multiple scales](#), *SIAM J. Math. Anal.*, **50** (2018), 1790–1816.
- [42] M. I. Jordan and T. M. Mitchell, [Machine learning: Trends, perspectives, and prospects](#), *Science*, **349** (2015), 255–260.
- [43] A. V. Joshi, *Machine learning and artificial intelligence*, Springer, Cham, 2020.
- [44] H. M. D. Kabir, A. Khosravi, M. A. Hosen and S. Nahavandi, [Neural network-based uncertainty quantification: A survey of methodologies and applications](#), *IEEE Access*, **6** (2018), 36218–36234.
- [45] C. Klingenberg, G. Puppo and M. Semplice, [Arbitrary order finite volume well-balanced schemes for the Euler equations with gravity](#), *SIAM J. Sci. Comput.*, **41** (2019), A695–A721.
- [46] N. B. Kovachki and A. M. Stuart, [Ensemble Kalman inversion: A derivative-free technique for machine learning tasks](#), *Inverse Probl.*, **35** (2019), 095005, 35 pp.
- [47] J. Kusch and M. Frank, [Intrusive methods in uncertainty quantification and their connection to kinetic theory](#), *Int. J. Adv. Eng. Sci. Appl. Math.*, **10** (2018), 54–69.
- [48] O. P. Le Maître and O. M. Knio, *Spectral Methods for Uncertainty Quantification*, With applications to computational fluid dynamics. Scientific Computation. Springer, New York, 2010.
- [49] C. Li and S. Mahadevan, [Efficient approximate inference in Bayesian networks with continuous variables](#), *Reliability Engineering & System Safety*, **169** (2018), 269–280.
- [50] C.-L. Li, W.-C. Chang, Y. Cheng, Y. Yang and B. Póczos, Mmd gan: Towards deeper understanding of moment matching network, in *Advances in Neural Information Processing Systems*, (2017), 2203–2213.
- [51] H. Lin and S. Jegelka, Resnet with one-neuron hidden layers is a universal approximator, in *Advances in Neural Information Processing Systems* (eds. S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi and R. Garnett), Curran Associates, Inc., **31** (2018), 6169–6178.
- [52] S. Mandt, M. Hoffman and D. Blei, Stochastic gradient descent as approximate Bayesian inference, *J. Mach. Learn. Res.*, **18** (2017), Paper No. 134, 35 pp.

- [53] W. S. McCulloch and W. Pitts, [A logical calculus of the ideas immanent in nervous activity](#), *Bull. Math. Biophys.*, **5** (1943), 115–133.
- [54] S. Mishra, Ch. Schwab and J. Šukys, [Multi-level Monte Carlo finite volume methods for nonlinear systems of conservation laws in multi-dimensions](#), *J. Comput. Phys.*, **231** (2012), 3365–3388.
- [55] S. Mishra, Ch. Schwab and J. Šukys, [Multilevel Monte Carlo finite volume methods for shallow water equations with uncertain topography in multi-dimensions](#), *SIAM J. Sci. Comput.*, **34** (2012), B761–B784.
- [56] O. Nikodym, [Sur une généralisation des intégrales de M. J. Radon](#), *Fundamenta Mathematicae*, **15** (1930), 131–179.
- [57] S. C. Onar, A. Ustundag, Ç. Kadaifci and B. Oztaysi, [The changing role of engineering education in industry 4.0 era](#), in *Industry 4.0: Managing The Digital Transformation*, Springer, (2018), 137–151.
- [58] P. Pettersson, G. Iaccarino and J. Nordström, [A stochastic Galerkin method for the Euler equations with Roe variable transformation](#), *Journal of Computational Physics*, **257** (2014), 481–500.
- [59] G. Poëtte, B. Després and D. Lucor, [Uncertainty quantification for systems of conservation laws](#), *J. Comput. Phys.*, **228** (2009), 2443–2467.
- [60] R. Pulch and E. Maten, [Stochastic galerkin methods and model order reduction for linear dynamical systems](#), *International Journal for Uncertainty Quantification*, **5** (2015), 255–273.
- [61] R. Pulch and D. Xiu, [Generalised polynomial chaos for a class of linear conservation laws](#), *J. Sci. Comput.*, **51** (2012), 293–312.
- [62] D. Ray and J. S. Hesthaven, [An artificial neural network as a troubled-cell indicator](#), *J. Comput. Phys.*, **367** (2018), 166–191.
- [63] D. Ray and J. S. Hesthaven, [Detecting troubled-cells on two-dimensional unstructured grids using a neural network](#), *J. Comput. Phys.*, **397** (2019), 108845, 31 pp.
- [64] N. H. Risebro, C. Schwab and F. Weber, [Multilevel Monte Carlo front-tracking for random scalar conservation laws](#), *BIT*, **56** (2016), 263–292.
- [65] L. Ruthotto, S. Osher, W. Li, L. Nurbekyan and S. W. Fung, [A machine learning framework for solving high-dimensional mean field game and mean field control problems](#), *Proceedings of the National Academy of Sciences*, **117**, (2020), 9183–9193.
- [66] L. Ruthotto and E. Haber, [Deep neural networks motivated by partial differential equations](#), *J. Math. Imaging Vision*, **62** (2020), 352–364.
- [67] G. Scarcioffi and A. Astolfi, [Data-driven model reduction by moment matching for linear and nonlinear systems](#), *Automatica J. IFAC*, **79** (2017), 340–351.
- [68] R. Schmitt and G. Schuh, [Advances in Production Research: Proceedings of the 8th Congress of the German Academic Association for Production Technology \(WGP\), Aachen, November 19-20, 2018](#), Springer, 2018.
- [69] C. Schwab and S. Tokareva, [High order approximation of probabilistic shock profiles in hyperbolic conservation laws with uncertain initial data](#), *ESAIM Math. Model. Numer. Anal.*, **47** (2013), 807–835.
- [70] M. Semplice, A. Coco and G. Russo, [Adaptive mesh refinement for hyperbolic systems based on third-order compact WENO reconstruction](#), *J. Sci. Comput.*, **66** (2016), 692–724.
- [71] R. Shu, J. Hu and S. Jin, [A stochastic Galerkin method for the Boltzmann equation with multi-dimensional random inputs using sparse wavelet bases](#), *Numer. Math. Theory Methods Appl.*, **10** (2017), 465–488.
- [72] R. C. Smith, *Uncertainty Quantification: Theory, Implementation, and Applications*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2013.
- [73] R. C. Smith, *Uncertainty Quantification: Theory, Implementation, and Applications*, Computational Science & Engineering, 12. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2014.
- [74] R. J. Solomonoff, *Machine Learning – Past and Future*, The Dartmouth Artificial Intelligence Conference, 2006.
- [75] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, 3rd edition, Texts in Applied Mathematics, 12. Springer-Verlag New York, 2002.
- [76] J. Šukys, S. Mishra and C. Schwab, [Multi-level Monte Carlo finite difference and finite volume methods for stochastic linear hyperbolic systems](#), *Springer Proceedings in Mathematics and Statistics*, **65** (2013), 649–666.

- [77] T. Tang and T. Zhou, [Convergence analysis for stochastic collocation methods to scalar hyperbolic equations with a random wave speed](#), *Commun. Comput. Phys.*, **8** (2010), 226–248.
- [78] H. Tercan, T. Al Khawli, U. Eppelt, C. Büscher, T. Meisen and S. Jeschke, [Improving the laser cutting process design by machine learning techniques](#), *Production Engineering*, **11** (2017), 195–203.
- [79] H. H. Thodberg, A review of bayesian neural networks with an application to near infrared spectroscopy, *IEEE Transactions on Neural Networks*, **7** (1996), 56–72.
- [80] S. Tokareva, C. Schwab and S. Mishra, High order SFV and mixed SDG/FV methods for the uncertainty quantification in multidimensional conservation laws, in *High Order Nonlinear Numerical Schemes for Evolutionary PDEs*, Lect. Notes Comput. Sci. Eng., Springer, Cham, **99** (2014), 109–133.
- [81] R. K. Tripathy and I. Bilionis, [Deep UQ: Learning deep neural network surrogate models for high dimensional uncertainty quantification](#), *J. Comput. Phys.*, **375** (2018), 565–588.
- [82] V. Vanhoucke, A. Senior and M. Z. Mao, [Improving the Speed of Neural Networks on CPUs](#). Deep Learning and Unsupervised Feature Learning Workshop, 2011.
- [83] Q. Wang, J. S. Hesthaven and D. Ray, [Non-intrusive reduced order modelling of unsteady flows using artificial neural networks with application to a combustion problem](#), *J. Comput. Phys.*, **384** (2019), 289–307.
- [84] K. Watanabe and S. G. Tzafestas, [Learning algorithms for neural networks with the Kalman filters](#), *J. Intell. Robot. Syst.*, **3** (1990), 305–319.
- [85] N. Wiener, [The homogeneous chaos](#), *Amer. J. Math.*, **60** (1938), 897–936.
- [86] D. Xiu, *Numerical methods for stochastic computations*, A spectral method approach. Princeton University Press, Princeton, 2010.
- [87] D. Xiu and G. E. Karniadakis, [The Wiener-Askey polynomial chaos for stochastic differential equations](#), *SIAM J. Sci. Comput.*, **24** (2002), 619–644.
- [88] M. Zanella, [Structure preserving stochastic Galerkin methods for Fokker-Planck equations with background interactions](#), *Math. Comput. Simulation*, **168** (2020), 28–47.
- [89] Y. Zhu and S. Jin, [The Vlasov-Poisson-Fokker-Planck system with uncertainty and a one-dimensional asymptotic preserving method](#), *Multiscale Model. Simul.*, **15** (2017), 1502–1529.

E-mail address: trimborn@igpm.rwth-aachen.de

E-mail address: gerster@igpm.rwth-aachen.de

E-mail address: visconti@igpm.rwth-aachen.de