GenUQ: Predictive Uncertainty Estimates via Generative Hyper-Networks

Tian Yu Yen
Sandia National Laboratories
Albuquerque, NM 87123
tyen@sandia.gov

Reese E. Jones
Sandia National Laboratories
Livermore, CA 94551
rjones@sandia.gov

Ravi G. Patel*
Sandia National Laboratories
Albuquerque, NM 87123
rgpatel@sandia.gov

Abstract

Operator learning is a recently developed generalization of regression to mappings between functions. It promises to drastically reduce expensive numerical integration of PDEs to fast evaluations of mappings between functional states of a system, i.e., surrogate and reduced-order modeling. Operator learning has already found applications in several areas such as modeling sea ice, combustion, and atmospheric physics. Recent approaches towards integrating uncertainty quantification into the operator models have relied on likelihood based methods to infer parameter distributions from noisy data. However, stochastic operators may yield actions from which a likelihood is difficult or impossible to construct. In this paper, we introduce, GenUQ, a measure-theoretic approach to UQ that avoids constructing a likelihood by introducing a generative hyper-network model that produces parameter distributions consistent with observed data. We demonstrate that GenUQ outperforms other UQ methods in three example problems, recovering a manufactured operator, learning the solution operator to a stochastic elliptic PDE, and modeling the failure location of porous steel under tension.

1 Introduction

Over the last few years there has been tremendous growth at the intersection of scientific machine learning (SciML), traditional scientific computing methods, and computer science algorithms. In particular, there has been growing interest in utilizing neural networks to learn physical operators of dynamical systems, such as high-dimensional partial differential equations (PDEs) Patel and Desjardins [2018], Patel et al. [2021], Lu et al. [2022a], Kovachki et al. [2023]. These methods are referred to under the general category of operator learning. However, in order for these operator learning approaches to achieve significant impact in scientific and engineering applications, they

*Corresponding author

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Structured Probabilistic Inference & Generative Modeling.

must have reliable and robust methods for assessing the uncertainties associated with training and deploying models.

Uncertainties from machine learning models are often placed into two categories: aleatoric uncertainties arising from inherent stochasticities, e.g., in stochastic ODEs and PDEs; and epistemic uncertainties arising from lack of data. Ref. Psaros et al. [2023] provides a thorough review of the types of uncertainties that can influence solutions to various dynamical systems obtained via SciML methods, such as operator learning. In this paper, we focus on the case of aleatoric uncertainties. In particular, we are interested in learning an operator that accurately estimates predictive uncertainty due to such variability. That is, given a fixed input x, what is the probability distribution p that best describes the action of a stochastic operator $\mathcal{M}$ acting on x?

Many current approaches, including Bayesian and Frequentist frameworks, rely on a likelihood function to model aleatoric uncertainty. The stochastic operator $\mathcal{M}$ is parameterized by some model architecture. Optionally, a prior probability distribution can also be placed on the parameters of the model. By assuming a form for the likelihood function on the data-model relationship, the parameters of the model can be found via maximum likelihood estimation (MLE) or by maximum a posteriori (MAP) estimation Psaros et al. [2023]. Often, the likelihood is also parameterized and optimized to more flexibly model the stochastic behavior of the system Kendall and Gal [2017]. These sample parameters can then be utilized to evaluate the model and, when convolved with the likelihood function, be used to estimate the predictive uncertainty of the output of the operator. However, these approaches can be computationally prohibitive to execute, and approximation methods are often used, such as assuming no correlation between model outputs Kendall and Gal [2017]. In addition, the computational complexity of likelihood based approaches, they often also require strong assumptions about the likelihood function which may be inappropriate for assessing the type of uncertainties relevant to systems of stochastic ODEs and PDEs.

Recently, generative modeling has demonstrated success across a variety of machine learning tasks Bengesi et al. [2024]. These models can also be interpreted as providing uncertainty quantification Mimikos-Stamatopoulos et al. [2024]. Score-based approaches towards generative modeling are particularly attractive because they neither require a likelihood nor involve adversarial optimization Pacchiardi and Dutta [2022]. Authors have used generative models as hyper-networks to generate the weights for a neural network model, allowing one to estimate uncertainties for model predictions Krueger et al. [2017], Ratzlaff and Fuxin [2019], Safta et al. [2025]. These works, however, did not provide isolated analysis of quantified aleatoric uncertainties apart from epistemic uncertainties. Moreover, instead of using score based approaches to generative modeling, they relied either on generative adversarial approaches or methods that require likelihood functions. Finally, none of these works have applied these hyper-network UQ approaches towards operator learning.

In this paper, we propose a new method of determining predictive uncertainty estimates for learned operators. We will refer to this method as GenUQ. Our approach is to determine a set of operators whose action is consistent with the distribution of observed data by training a generative hyper-network to sample a corresponding distribution of model parameters. The generative hyper-network can be trained at the same time as a traditional operator learning model using standard backpropagation. In addition, the approach requires no assumptions about the likelihood of the data nor a prior distribution on model parameters (though an appropriate prior initialization may benefit training). Surprisingly, though the parameterization of models may be high-dimensional, we show the generative hyper-network need only produce a small subset of these to produce an appropriate distribution of predictions. Therefore, GenUQ is only marginally more expensive to train and perform inference on than a deterministic model. Though our interest is on applications in operator learning, the approach is general enough to be applied to other types of machine learning problems.

2 GenUQ: Predictive Uncertainty from Generative Models

Suppose we have a dataset consisting of N pairs of data, $(u_i, v_i)^{i\in 1,2,\ldots,N} \in \mathcal{U} \times \mathcal{V}$, and wish to model it by providing a function (e.g., a neural network), $f : \mathcal{U} \times \Theta \to \mathcal{V}$ with appropriate parameters, $\theta \in \Theta$. In our scenario, the phenomenon that produced the dataset is stochastic in the sense that repeated experiments on the same input, u, from the probability measure, $\mathbb{P}_{\mathcal{U}}(U)$, would yield outputs from a measure, $v \sim \mathbb{P}_{\mathcal{V}}(V|U = u)$. For our applications, $\mathcal{U}$ and $\mathcal{V}$ will be subspaces of L^2 with norms, $||\cdot||_{\mathcal{U}}$ and $||\cdot||_{\mathcal{V}}$. We are particularly interested in operator learning based PDE surrogate

models, so these spaces will be of functions over a domain, $\Omega \subset \mathbb{R}^d$. Additionally, we will not assume that we have access to multiple realizations of the phenomenon for the same input, i.e., the dataset consists of unique u_i's.

Similar to Bayesian frameworks such as Neal [2012], Meng et al. [2021], we assume that the parameters θ are random variables and can model the dataset by providing a probability measure for $\mathbb{P}_\Theta$ or a means of sampling $\theta \sim \mathbb{P}_\Theta$ such that its pushforward matches the data. However, instead of following the traditional frameworks of finding a distribution for θ given the data through Bayes' Rule, we instead seek to estimate a measure $\mathbb{P}_\Theta$ whose push-forward approximates the output target measure $\mathbb{P}_\mathcal{W}(W)$. That is,

$$\mathbb{P}_\Theta\left(f_u^{-1}(V)\right) \equiv \widehat{\mathbb{P}}_\mathcal{V}(V|U=u) \approx \mathbb{P}_\mathcal{V}(V|U=u) \tag{1}$$

where $f_u^{-1}(V) = \{\theta \in \Theta : f_\theta(u) = V\}$. This approximation should hold over any $u \sim \mathbb{P}_\mathcal{U}$, so we seek a model that approximates the joint distribution, $\widehat{\mathbb{P}}_\mathcal{W}(W) \approx \mathbb{P}_\mathcal{W}(W)$, where $\mathcal{W} \equiv \mathcal{U} \times \mathcal{V}$ is also a L^2 space whose elements consist of $h_u(\theta) = (u, f_\theta(u))$. Therefore, we require consistency between the pushforward of our parameter measure and the data distribution,

$$\mathbb{P}_\Theta\left(h^{-1}(W)\right) \equiv \widehat{\mathbb{P}}_\mathcal{W}(W) \approx \mathbb{P}_\mathcal{W}(W) \tag{2}$$

where $h^{-1}(W) = \{\theta \in \Theta : h_U(\theta) = (U, f_\theta(U)) = W\}$. This approach bears similarity to other measure-theoretic frameworks for uncertainty quantification Butler et al. [2018, 2025].

Note that in many applications, the uncertainty in parameters θ of a black-box model is not of primary interest: what is of interest is the accuracy and appropriateness of the approximate predictive measure $\widehat{\mathbb{P}}_\mathcal{W}$. Therefore, we propose to use an hyper-network generative model g_ϕ to learn to sample from the measure $\mathbb{P}_\theta$. Let $z \sim \mathbb{P}_\mathcal{Z}$ be from a measure that is straightforward to sample, such as the standard normal distribution. We train a simple, generative feed-forward network $g_\phi : \mathcal{Z} \to \Theta$ such that

$$g_\phi(z) = \theta \;\rightarrow\; f_\theta(u) = \hat{v}$$

and we minimize the differences between the resulting distribution of $\hat{w} = (u, \hat{v})$ and $w = (u, v)$ using a discrepancy function D:

$$\min_\phi D(\widehat{\mathbb{P}}_\mathcal{W}, \mathbb{P}_\mathcal{W}). \tag{3}$$

As discussed in the following section, there are many options for learning such a sampling distribution, including normalizing flows Rezende and Mohamed [2015a]. In our experiments, we find that utilizing a non-invertible feed-forward neural network for g_ϕ is sufficient provided an appropriate discrepancy function. To maintain the computational efficiency of GenUQ, we seek a generative model for only a small, random subset of the underlying model's parameters, leaving the remainder as deterministic variables. The proportion of generated parameters, R, is a hyperparameter that we examine in the results section.

There are many metrics and discrepancy scores D that could be utilized to estimate the differences between probability measures $\widehat{\mathbb{P}}_\mathcal{W}$ and $\mathbb{P}_\mathcal{W}$, such as the KL-divergence and the negative log likelihood. We choose to utilize the energy score which is a strictly proper score in the sense that it obtains a minimum if and only if $\widehat{\mathbb{P}}_\mathcal{W} = \mathbb{P}_\mathcal{W}$,

$$\begin{aligned} D_e(\widehat{\mathbb{P}}, \mathbb{P}) &:= -\frac{1}{2}\,\mathbb{E}_{w\sim\hat{\mathbb{P}}, w'\sim\mathbb{P}}\left[\rho(w,w')\right] + \mathbb{E}_{w,w'\sim\hat{\mathbb{P}}}\left[\rho(w,w')\right] \\ \rho(w,w') &= \|w-w'\|_\mathcal{W}^\beta = \left(\|u-u'\|_\mathcal{U}^2 + \|v-v'\|_\mathcal{V}^2\right)^{\beta/2} \end{aligned} \tag{4}$$

where $\beta \in (0,2)$. Note that the two terms in the norm, $\|\cdot\|_\mathcal{W}$ can be weighted if $\|u\|_\mathcal{U}/\|v\|_\mathcal{V} \neq \mathcal{O}(1)$. This score is associated with the maximum-mean discrepancy (MMD) metric for a specific choice of kernel distance Bińkowski et al. [2018]. There are a number of advantages to utilizing such a discrepancy score to train models. See Gneiting and Raftery [2007], Pacchiardi and Dutta [2022] for details. In particular, one of the advantages of using this type of discrepancy score is that explicit knowledge of the underlying distributions is not required to approximate the energy score (as with other MMD metrics), i.e., the energy score can be approximated via Monte Carlo from finite samples of the two comparison distributions $\mathbb{P}$ and $\mathbb{Q}$. Indeed, we have samples from the joint distribution

$\{w_i = (u_i, v_i)\} \sim \mathbb{P}_{\mathcal{W}}$ in practice. Thus, we can train the hyper-network generative model with the following criteria:

$z_{ij} \sim \mathbb{P}_{\mathcal{Z}}$:	samples from generating distribution
$\theta_{ij} = g_\phi(z_{ij})$:	model parameter samples
$\hat{v}_{ij} = f_{\theta_{ij}}(u_i)$:	model outputs
$\hat{w}_{ij} = (u_i, \hat{v}_{ij})$:	predicted samples
$\min_\phi \widetilde{D}_e(\widehat{\mathbb{P}}_{\mathcal{W}}, \mathbb{P}_{\mathcal{W}})$:	objective function

where

$$\widetilde{D}_e(\widehat{\mathbb{P}}_{\mathcal{W}}, \mathbb{P}_{\mathcal{W}}) = -\frac{1}{2mn} \sum_i \sum_j \rho\left(\hat{w}_{ij}, w_i\right) + \frac{1}{m(n-1)} \sum_i \sum_{j \neq j'} \rho\left(\hat{w}_{ij}, \hat{w}_{ij'}\right).$$

where m the number of samples in the training set and n_z is the number of model predictions and a tunable optimization hyperparameter. Although we focus on operator learning examples in the results section, we note that GenUQ is not limited to operator learning and is applicable to other regression tasks, e.g., $\mathcal{U} = \mathbb{R}^n$ and $\mathcal{V} = \mathbb{R}^m$.

3 Related Works

Bayesian approaches for uncertainty quantification are very popular, especially in the scientific machine learning community. Among them, variational inference is commonly chosen because it enables inference for posterior parameters in situations where performing MCMC sampling is challenging Graves [2011], Blei et al. [2017]. Nonetheless, classical variational inference approaches can be limited by assumptions on the class of approximation densities used to approximate the posterior (e.g., mean-field approximations). To overcome this limitation, strategies such as normalizing flows have been proposed to generate a more flexible class of approximation densities for use in the variational inference approach Rezende and Mohamed [2015b]. Our approach is in a similar spirit to this type of approach but differs in the mathematical framework underlying the approach. While variational inference also seeks a distribution over model parameters, the posterior distribution it attempts to approximate is assumed to be a convolution of the data likelihood and prior distribution. Instead, we seek a distribution over the model parameters whose push-forward measure is consistent with the distribution of the data, i.e., via the discrepancy score (4). These differences in objectives can lead to significantly different mathematical consequences–see Butler et al. [2018] for details. Our approach is most similar to Lingsch et al. [2024], which learns a solution to an inverse problem (in addition to the forward operator) to characterize the stochastic inputs to a PDE. They associate every input $u \in \mathcal{U}$ with a specific stochastic parameterization $\xi \mid u \in \Xi$, and then learn the appropriate mapping $f_\theta : \Xi \to \mathcal{V}$. On the other hand, we assume that ξ is independent of u and directly influences the output v through the operator. Thus, our method, GenUQ, focuses on a forward problem directly relating $\mathcal{U}$ and $\mathcal{V}$ (via $f_\theta : \mathcal{U} \to \mathcal{V}$), bypassing the need to solve an extra inverse problem.

4 Results

We evaluate and compare GenUQ to other UQ methods in three examples. Our first two examples utilize neural operators as the underlying model, f, while the final example uses a convolutional neural network. For each example, we have a dataset of input/output function pairs (u, v), that we partition into training/validation/test sets with a 60%/20%/20% split. We train all models with the Adam optimizer using a stepped learning rate schedule: $(10^{-3}, 10^{-4}, 10^{-5}, 10^{-6})$, with 400 epochs per learning rate. We track the validation loss over the training process and report the model with the lowest validation loss. All experiments were performed on a 80GB A100 NVIDIA GPU.

Throughout these examples, we compare to 5 other UQ methods. The first three are maximum likelihood estimation based methods where we assume Gaussian error with heteroscadastic, but uncorrelated noise in a mean field variational inference model (VI), heteroscadastic noise with full covariance structure (CoV), and normalizing flow conditioned on the input function (NF). While VI provides both epistemic and aleatoric uncertainties, we include it due to its ubiquity in machine

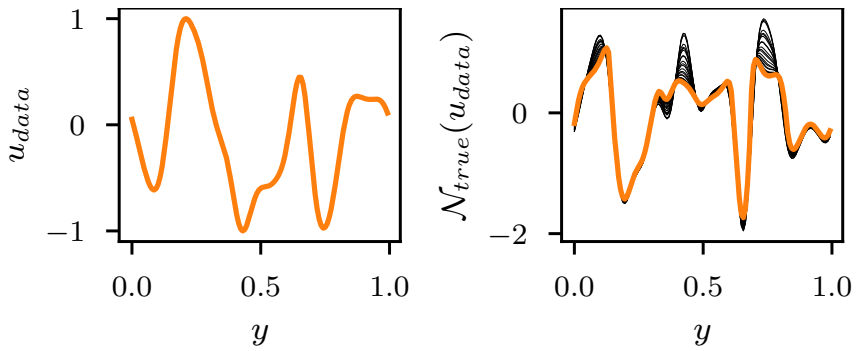


Figure 1: *(Left)* Sample input function in training data. *(Right, orange)* Sample output function in training data. *(Right, black)* Actions of resampled operators on same input function.

learning. For CoV, we predict a full covariance matrix from the neural operator. For NF, we use masked autoregressive flow Papamakarios et al. [2017] from the error distribution to a normal distribution using the FlowJax library with default options Ward [2025].

The forth and fifth methods we compare to GenUQ are a dropout (DO) architecture and a generative model that lacks the hyper-network and maps the input function concatenated with noise to the stochastic output (Generative). Both architectures are trained using the same energy score in (4)

4.1 ELU Operator learning example

We compare GenUQ to three other UQ methods in recovering a manufactured stochastic operator from its action on random smooth functions. The operator we use is the composition between the exponential linear unit (ELU) point-wise non-linearity Clevert et al. [2016] and a differentiation,

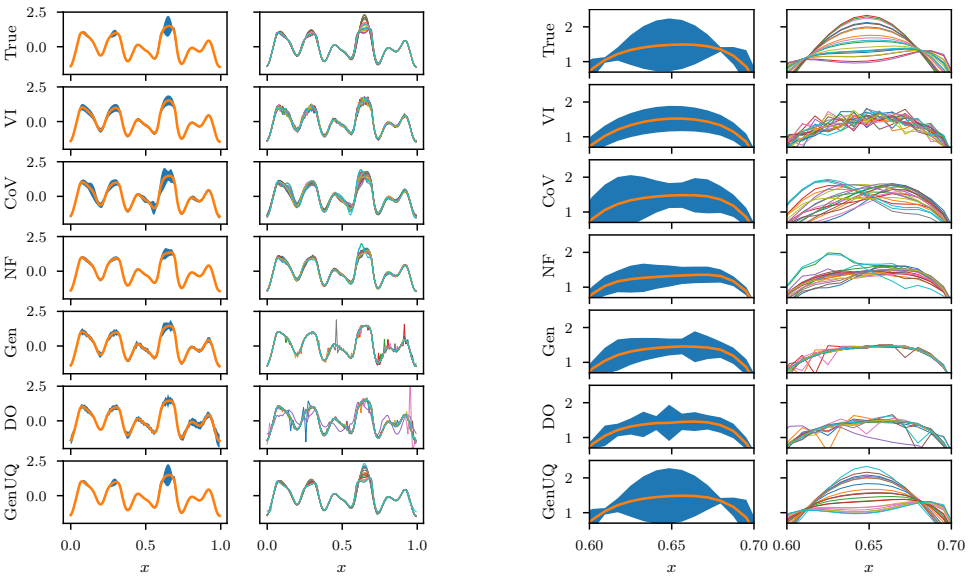
$$\begin{aligned} &\mathcal{N}(u) = \partial_x f \\ &f(x) = \mathrm{ELU}(u(x) + \alpha) - \alpha \\ &\alpha \sim U[0, 1] \end{aligned} \tag{5}$$

To generate a set of input functions, u_i, we generate smooth random functions on a periodic 1D domain, $[0, 2\pi]$, by sampling a Gaussian process with a mean of zero and a covariance kernel, $K(x, x') = \exp(4\cos(2\pi(x - x')))$. For each input function, u_i, we sample an operator, $\mathcal{N}_i$, as per (5), and apply it to u_i to generate $v_i = \mathcal{N}_i(v_i)$. We try to recover the stochastic operator in (5) using function pairs $\{u_i, v_i\}^{i=1,\dots,N}$. Our full dataset consists of $N = 2048$ sample pairs. In Figure 1 we show a pair of input and output functions. Additionally, we show 20 samples of the action of the stochastic operator on the same input function to qualitatively show the stochasticity of the operator.

Our underlying model is the operator learning method, Modal Operator Regression for Physics (MOR-Physics)Patel and Desjardins [2018], Patel et al. [2021], $f_\theta(u) = \mathcal{F}^{-1} g_{\theta_g}(\kappa)\mathcal{F} h_{\theta_h}(u)$, where $\mathcal{F}$ is the Fourier transform, κ is the wavenumber, and g and h are neural networks. For both neural networks, we use a depth of 6 and width of 32 and the GELU activation function. For the generator network, we use a four layer network of widths, $(10, 20, 30, 40)$, and the same activation function.

In addition to training a GenUQ model on this dataset, we train the five other UQ methods discussed above. After training the models, we generate new input functions from the same Gaussian process to compare model performance. Figure 2 shows the comparison between the true operator (1), the VI, CoV, NF, DO, Gen, and GenUQ operators. We find that the GenUQ operator better captures the stochasticity of the true operator than the other methods and has lower energy distance. The 95% confidence intervals provided by GenUQ more closely follow those of the true operator compared to than the others. Moreover, the actions for the GenUQ operator remain smooth while the error model in VI that assumes heteroscadastic but independent Gaussian noise loses the correlation structure found in the data. Similarly, Gen and DO both produce noisy predictions. CoV and NF is able to capture smooth functions but the Gaussian restriction for CoV and optimization challenges for both methods led to to poor quality UQ.

In Figure 3a, we look at the distribution of the action, v, at three different points in the domain. For clarity, we only compare GenUQ to VI. We observe that GenUQ produces distributions that are closer



(a) Full input range

(b) Focused input range

Figure 2: Comparison of UQ methods for recovering the 1D operator in (5). *(Left)* the predictive mean *(orange)* and 95% confidence intervals *(blue)* of the action of each method on a test function. *(Right)* Actions of resampled operators on same test function. Qualitatively, the learned GenUQ operator better matches the true operator. Energy distances between predictions and data, VI: 0.0265, CoV: 78.4386, NF: 0.1215, Gen: 0.2122, DO: 0.1994, GenUQ: 0.0020.

to the data than VI. In Figure 3b, we look more closely at the correlation structure for the actions of the three operators by plotting sample points, $(v_t(x_0), v_t(x_1))$ for $x_0 = 0.6$ and varying x_1. We find that VI has weak correlations throughout while GenUQ matches closely the correlation structure of the true operator. Additionally, we plot the histograms of $v_t(x_1)$ for the three operators for varying x_1 and find GenUQ again to more closely match the true operator than VI. GenUQ, however, does not exactly match the true operator and produces distributions for the actions that are more diffuse than the sharp distributions from the true operator.

We assess the sensitivity of the GenUQ model to the random generator dimension d and the number of model parameters treated as stochastic S. We measured the quality of the predicted distributions for each d and S using the energy distance, which is equivalent to the energy score (4) plus $\mathbb{E}_{w,w'\sim\mathbb{P}}\left[\rho(w, w')\right]$. We also checked the accuracy of the predicted sample means in the L^2-norm. Table 1 shows the results. For this example, only a small proportion of the total number of parameters needs to be utilized to reproduce reasonable actions of the learned stochastic operator. This is very likely due to the fact that the stochasticity of the true operator is governed by a single random variable, α. On the other hand, treating too many model parameters as stochastic leads to a lack of convergence. We hypothesize that the model needs to be reasonably calibrated to the data in order for the model to converge to an appropriate solution. When too many parameters are stochastic, the non-uniqueness of model parameters leads to degeneracy of the learning process.

4.2 Stochastic Poisson equation solution operator

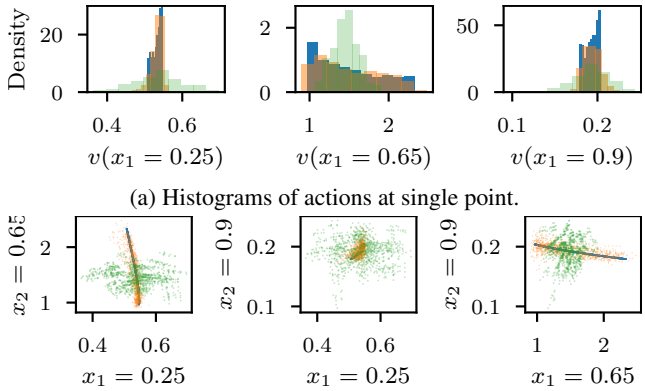
In our next example, we compare GenUQ to VI for recovering the solution operator for a stochastic nonlinear Poisson equation. We consider the following PDE on the unit disk in 2D,

$$\begin{aligned} \nabla \cdot a(\nabla v) &= u \quad && ||x||_2 < 1 \\ u &= 0 && ||x||_2 = 1 \\ a \sim \mathcal{A} \quad u &\sim \mathcal{U} \end{aligned} \tag{6}$$

We select for $\mathcal{U}$ a Gaussian process, $\mathcal{G}[0, K(x, x') = 0.04\exp(6.25||x - x'||_2^2)]$. To maintain the well-posedness of the PDE, $a : \mathbb{R}^2 \to \mathbb{R}^2$ must be a monotone function [Evans, 2022, Chapter 9], so $\mathcal{A}$ must be a distribution over monotone functions. We do not explicitly construct $\mathcal{A}$, but instead

$S = (\%$ of $M)$	$d = (\%$ of $S)$			
	25	50	75	100
0.1	0.005 (2.2e-4)	0.004 (2.0e-4)	**0.002 (1.6e-4)**	0.005 (2.1e-4)
0.4	0.003 (1.9e-4)	0.004 (1.9e-4)	0.003 (1.8e-4)	0.002 (1.7e-4)
1.6	0.005 (5.2e-4)	0.005 (1.9e-4)	0.003 (1.7e-4)	0.008 (2.8e-4)
6.4	0.122	DnC	DnC	DnC
25.6	DnC	DnC	DnC	DnC

Table 1: Shows the GenUQ model performance for different numbers of stochastic parameters S (as a proportion of M) and generating dimension d (as a proportion of S); the total number of model parameters was $M = 16,131$. Performance measured in the energy distance (4) and differences between sample means in L^2-norm in parentheses. DnC: did not converge.



(a) Histograms of actions at single point.

(b) Pair-wise values of actions.

Figure 3: Fine grained evaluation of GenUQ in ELU operator learning example. Stochastic operators' actions on test input function in Figure 2. *(blue)* Test data. *(green)* VI. *(orange)* GenUQ.

sample $\mathcal{A}$ by randomly initializing a one hidden layer of width 2 monotone neural network with hard sigmoid activations. To generate the training, validation, and test data, we numerically solve (6) using a finite element solver with P1 basis functions on a triangular mesh with characteristic mesh size, $h = 0.025$. We generate a dataset of 10000 samples. In Figure 4 we show sample input and output function pairs from the test dataset.

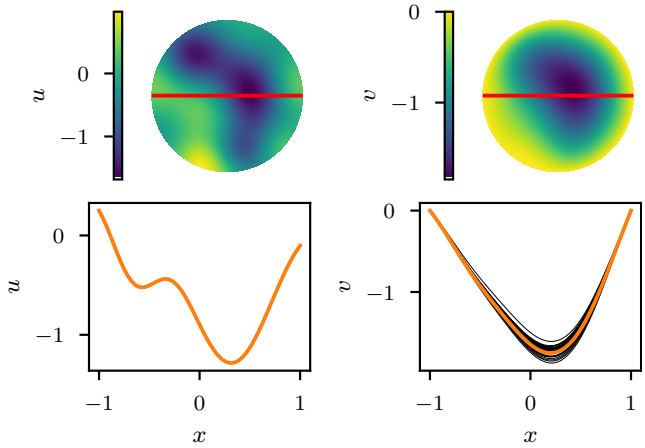
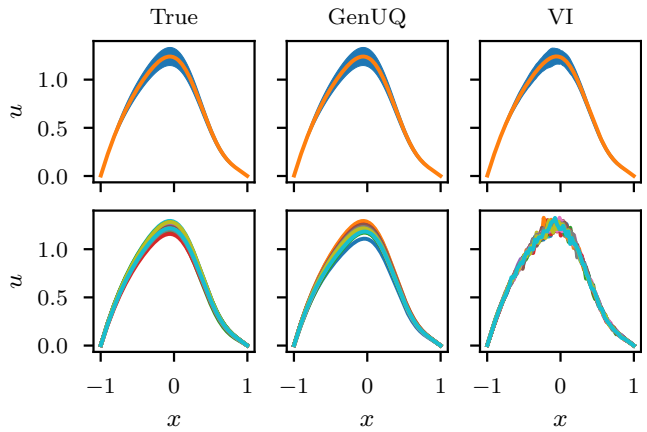
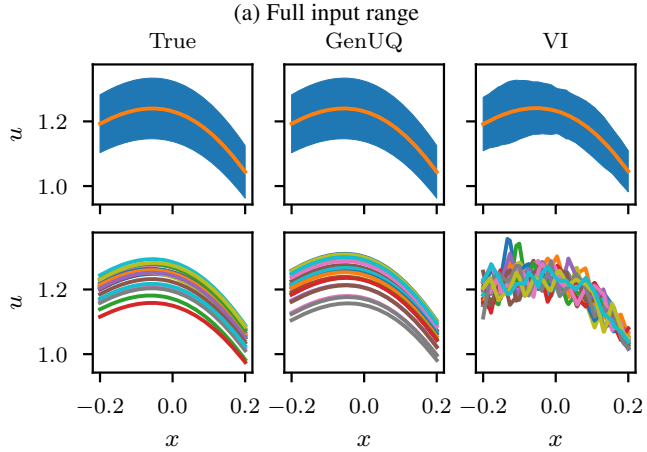


Figure 4: *(Top Left)* Sample input function from training set. *(Top Right)* Sample action on input function from training set. *(Bottom Right)* Trace of sample input function along the red line in the corresponding contour plot. *(Bottom Left, orange)* Trace of sample output function along the red line. *(Bottom Left, black)* Action of multiple realizations of solution operator on fixed input function.



(a) Full input range



(b) Focused input range

Figure 5: Visual comparison of VI and GenUQ for recovering the stochastic Poisson equation. between predictions and data, VI: 0.2643 and GenUQ: 0.0780

We will refer to the stochastic solution operator for (6) provided by the finite element solver as N_{true} and seek a neural operator, N, that emulates it. We parameterize the neural operator with the POD-DeepONets architecture with hard constrained boundary conditions Lu et al. [2022b]. We construct a truncated POD basis of dimension $d = 200$ from the training input functions, and project the input functions onto this basis to produce the POD coefficients, c. The POD-DeepONets architecture has the following form,

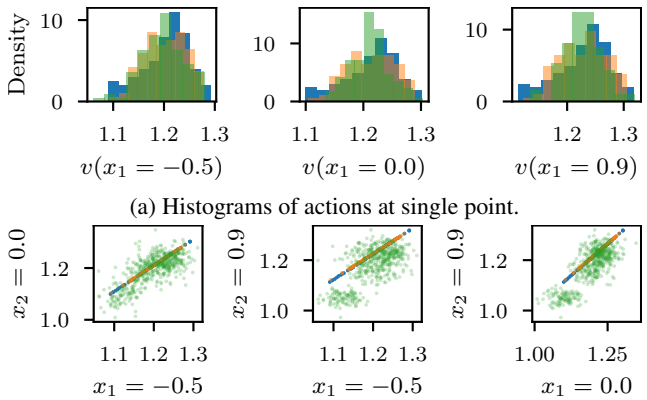
$$\begin{aligned} f_\theta(u)(x) &= g(x)(b_{\theta_b}(c) \cdot t_{\theta_t}(x)) \\ g(x) &= 1 - x^2 \end{aligned} \tag{7}$$

where t and b are trunk and branch neural networks of depth 5 and width 128. See Lu et al. [2022b] for further details on this architecture.

We train VI and GenUQ models with the same underlying architecture. In Figure 5, we evaluate both models on a test input function. As in the previous section, we find that GenUQ more closely matches the test data than VI. Again, VI produces noisy predictions due to the poorly specified likelihood function. The energy distance for the GenUQ model is also lower. In Figure 6a we plot histograms for the predictions of the three operators for an test input function at three points and find that GenUQ matches the data more closely than VI. Figure 6b, we find that the pair-wise values for GenUQ at these points also more closely matches the test data.

4.3 Porosity Example

For our last example, we apply GenUQ to learn the mapping between the porosity of a steel sample and the location of its failure under strain. We use high fidelity data from Khalil et al. [2021] which



(a) Histograms of actions at single point.

(b) Pair-wise values of actions.

Figure 6: Fine grained evaluation of GenUQ in Poisson example. Stochastic operators' actions on test input function in Figure 5. *(blue)* Test data. *(green)* VI. *(orange)* GenUQ.

	Deterministic	GenUQ				
		$n_z = 1$	$n_z = 4$	$n_z = 16$	$n_z = 64$	$n_z = 256$
Acc	64.24	63.28	77.35	84.56	89.84	93.41
ℓ_2	48.94	42.47	30.34	23.15	16.62	11.86

Table 2: Metrics for porosity damage problem. Acc refers to accuracy and ℓ_2 refers to relative ℓ_2 error. Both values are reported as percentages. For GenUQ, the best prediction out of n_z are reported. GenUQ outperforms the deterministic model even for $n_z = 1$.

consists of 0.75μm$\times 0.75\mu$m$\times 4\mu$m volumetric fields for the porosity of the material, u, and damage at failure, v. The damage field is in the range $[0, 1]$ where it takes a value of 1 at the point of failure. The porosity of the material was sampled from a Karhunen-Loeve process. For each porosity field, the authors conducted a finite element simulation of the material under strain until ductile failure. See Khalil et al. [2021] for further details on the simulation. The total number of simulations in the dataset is 9457 which we partition into training, test, and validation datasets.

A deterministic model for a mapping between porosity, u, and failure is challenging to obtain due to the extremely sensitive and highly localized nature of the map. Notably, the location of failure is often near a pore, but it is difficult to predict near which pore or cluster of pores the material will fail at. Additionally, the strain required break the material has high sample variance. Furthermore, each finite element simulation utilized approximately 16 CPU-hours. An accurate deterministic model would need to be inordinately complex to capture the behaviour in this dataset. Alternatively, we can recast the problem as one of predicting aleatoric uncertainty, i.e., given a sample porosity field, predict a distribution of failure locations. In this context, the true failure point would lie in a high density region in the support of a simpler, stochastic model's predictions.

We compare a deterministic and GenUQ model in predicting the failure location. For both models, we use the same underlying convolutional neural network (ConvNet) architecture to find a mapping between the porosity, u, and damage, v, training on pairs of fields from the training set. The ConvNet consists of 7 blocks of 3x3 convolution, batch normalization, and GELU activation. The convolutions in the blocks have filter sizes, (4,8,16,16,8,4,1). To the find failure location, we find the location of the maximum damage predicted. In Figure 7 in the appendix we show a sample pair of porosity and damage fields from the test set and corresponding damage predictions from the trained models. In Table 2, we compare statistics from test data predictions from the two models. Given enough samples from the generative model, it is able to successfully predict the failure location and damage field. The deterministic model, in contrast has very poor accuracy and high energy distance. Notably, even with only a single prediction, the generative model still outperforms the deterministic model. While this observation will require further investigation, we suspect that GenUQ may have a regularizing effect in addition to providing UQ.

5 Conclusion

In this work, we develop a UQ strategy for modeling aleatoric uncertainty with a focus on neural operators. Stochastic nonlinear operators can produce data with complicated error structures that are difficult to model with traditional likelihood based approaches. We develop an alternative approach that avoids constructing a likelihood by introducing an hyper-network generative model that produces model parameters consistent with the data distribution after training on an energy score. This approach outperforms standard UQ approaches in a variety on operator learning tasks. While our method is capable of quantifying aleatoric uncertainty for a variety of problems, it has limited capacity for quantifying epistemic uncertainty. Additionally, while the method was successful in the three examples we studied, it is unclear how well it will perform for other tasks. Future work will focus on extending this method to better quantify epistemic uncertainty and further applications.

Acknowledgment

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

SAND Number: SAND2025-12223C

References

Ravi G Patel and Olivier Desjardins. Nonlinear integro-differential operator regression with neural networks. *arXiv preprint arXiv:1810.08552*, 2018.

Ravi G. Patel, Nathaniel A. Trask, Mitchell A. Wood, and Eric C. Cyr. A physics-informed operator regression framework for extracting data-driven continuum models. *Computer Methods in Applied Mechanics and Engineering*, 373:113500, 2021.

Lu Lu, Xuhui Meng, Shengze Cai, Zhiping Mao, Somdatta Goswami, Zhongqiang Zhang, and George Em Karniadakis. A comprehensive and fair comparison of two neural operators (with practical extensions) based on fair data. *Computer Methods in Applied Mechanics and Engineering*, 393:114778, 2022a. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2022.114778. URL `https://www.sciencedirect.com/science/article/pii/S0045782522001207`.

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89):1–97, 2023. URL `http://jmlr.org/papers/v24/21-1524.html`.

Apostolos F. Psaros, Xuhui Meng, Zongren Zou, Ling Guo, and George Em Karniadakis. Uncertainty quantification in scientific machine learning: Methods, metrics, and comparisons. *Journal of Computational Physics*, 477:111902, 2023. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2022.111902. URL `https://www.sciencedirect.com/science/article/pii/S0021999122009652`.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 5580–5590, 2017.

Staphord Bengesi, Hoda El-Sayed, Md Kamruzzaman Sarker, Yao Houkpati, John Irungu, and Timothy Oladunni. Advancements in generative ai: A comprehensive review of gans, gpt, autoencoders, diffusion model, and transformers. *IEEe Access*, 12:69812–69837, 2024.

Nikiforos Mimikos-Stamatopoulos, Benjamin Zhang, and Markos Katsoulakis. Score-based generative models are provably robust: an uncertainty quantification perspective. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL `https://openreview.net/forum?id=KI5TANEO2e`.

Lorenzo Pacchiardi and Ritabrata Dutta. Score matched neural exponential families for likelihood-free inference. *Journal of Machine Learning Research*, 23(38):1–71, 2022.

David Krueger, Chin-Wei Huang, Riashat Islam, Ryan Turner, Alexandre Lacoste, and Aaron Courville. Bayesian hypernetworks. *arXiv preprint arXiv:1710.04759*, 2017.

Neale Ratzlaff and Li Fuxin. Hypergan: A generative model for diverse, performant neural networks. In *International Conference on Machine Learning*, pages 5361–5369. PMLR, 2019.

Cosmin Safta, Reese E Jones, Ravi G Patel, Raelynn Wonnacot, Dan S Bolintineanu, Craig M Hamel, and Sharlotte LB Kramer. Uncertainty quantification of neural network models of evolving processes via langevin sampling. *arXiv preprint arXiv:2504.14854*, 2025.

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

Xuhui Meng, Hessam Babaee, and George Em Karniadakis. Multi-fidelity bayesian neural networks: Algorithms and applications. *Journal of Computational Physics*, 438:110361, 2021.

T. Butler, J. Jakeman, and T. Wildey. Combining push-forward measures and bayes' rule to construct consistent solutions to stochastic inverse problems. *SIAM Journal on Scientific Computing*, 40(2):A984–A1011, 2018. doi: 10.1137/16M1087229. URL `https://doi.org/10.1137/16M1087229`.

Troy Butler, Rylan Spence, Timothy Wildey, and Tian Yu Yen. Stability and cconvergence of solutions to stochastic inverse problems using approximate probability densities. *International Journal for Uncertainty Quantification*, 15(4):21–51, 2025. ISSN 2152-5080.

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

Mikołaj Bińkowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=r1lUOzWCW`.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

Alex Graves. Practical variational inference for neural networks. *Advances in neural information processing systems*, 24, 2011.

David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe and. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017. doi: 10.1080/01621459.2017.1285773. URL `https://doi.org/10.1080/01621459.2017.1285773`.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1530–1538, Lille, France, 07–09 Jul 2015b. PMLR. URL `https://proceedings.mlr.press/v37/rezende15.html`.

Levi E. Lingsch, Dana Grund, Siddhartha Mishra, and Georgios Kissas. FUSE: Fast unified simulation and estimation for PDEs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL `https://openreview.net/forum?id=dbnEf790Kv`.

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

Daniel Ward. Flowjax: Distributions and normalizing flows in jax. 2025. doi: 10.5281/zenodo.10402073. URL `https://github.com/danielward27/flowjax`.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2016.

Lawrence C Evans. *Partial differential equations*, volume 19. American Mathematical Society, 2022.

Lu Lu, Xuhui Meng, Shengze Cai, Zhiping Mao, Somdatta Goswami, Zhongqiang Zhang, and George Em Karniadakis. A comprehensive and fair comparison of two neural operators (with practical extensions) based on fair data. *Computer Methods in Applied Mechanics and Engineering*, 393:114778, 2022b.

M. Khalil, G.H. Teichert, C. Alleman, N.M. Heckman, R.E. Jones, K. Garikipati, and B.L. Boyce. Modeling strength and failure variability due to porosity in additively manufactured metals. *Computer Methods in Applied Mechanics and Engineering*, 373:113471, 2021. ISSN 0045-7825.

A Qualitative evaluation of porosity fields predicted by GenUQ model

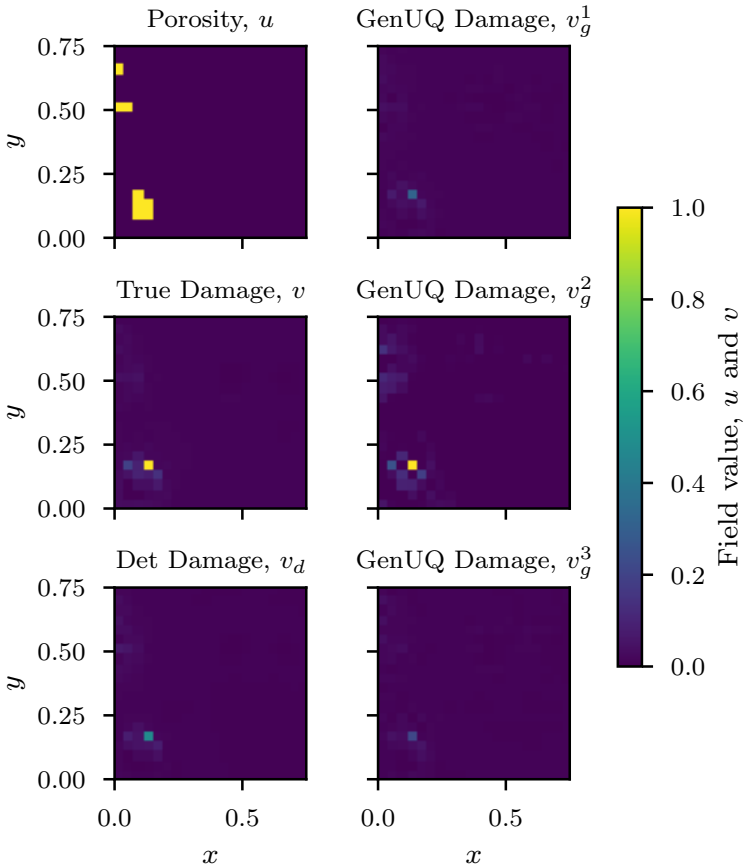


Figure 7: *(Top Left)* slice of a sample porosity field from the test set at $z = 2.35$ *(Middle Left)* Corresponding slice of damage field showing failure at $(x, y) = (0.1125, 0.15)$. *(Bottom Left)* Damage field predicted by deterministic model at same slice. Damage appears attenuated at this location. Failure point predicted incorrectly at $(x, y, z) = (0.0188, 0.3563, 0.3)$. *(Left)* Three samples of damage fields predicted by GenUQ. GenUQ will occasionally predict the correct damage behavior and failure location at the cost of also sampling erroneous behaviors.