



RESEARCH ARTICLE

# Neural network ensembles and uncertainty estimation for predictions of inelastic mechanical deformation using a finite element method-neural network approach

Guy L. Bergel<sup>1</sup> , David Montes de Oca Zapiain<sup>2</sup>  and Vicente Romero<sup>2</sup>

<sup>1</sup>Sandia National Laboratories, Livermore, CA, USA

<sup>2</sup>Sandia National Laboratories, Albuquerque, NM, USA

**Corresponding author:** Guy L. Bergel; Email: [bergel1@llnl.gov](mailto:bergel1@llnl.gov)

**Received:** 31 May 2023; **Revised:** 18 August 2023; **Accepted:** 20 September 2023

**Keywords:** Bayesian neural networks; finite element method; neural network ensembles; surrogate models; uncertainty quantification; variational Bayesian inference

## Abstract

The finite element method (FEM) is widely used to simulate a variety of physics phenomena. Approaches that integrate FEM with neural networks (NNs) are typically leveraged as an alternative to conducting expensive FEM simulations in order to reduce the computational cost without significantly sacrificing accuracy. However, these methods can produce biased predictions that deviate from those obtained with FEM, since these hybrid FEM-NN approaches rely on approximations trained using physically relevant quantities. In this work, an uncertainty estimation framework is introduced that leverages ensembles of Bayesian neural networks to produce diverse sets of predictions using a hybrid FEM-NN approach that approximates internal forces on a deforming solid body. The uncertainty estimator developed herein reliably infers upper bounds of bias/variance in the predictions for a wide range of interpolation and extrapolation cases using a three-element FEM-NN model of a bar undergoing plastic deformation. This proposed framework offers a powerful tool for assessing the reliability of physics-based surrogate models by establishing uncertainty estimates for predictions spanning a wide range of possible load cases.

## Impact Statement

This manuscript highlights one of the first approaches for quantifying the uncertainty in deep learning-based surrogate modeling coupled with finite-element analysis, which is a rising area of research and interest given their potential to offset the computational cost of finite element methods alone.

## 1. Introduction

Mechanical deformation of elastic materials is greatly influenced by microscale features (such as grains, thermodynamical phases, inclusions) as well as geometric features (such as cracks, notches, voids, etc.) (Knott, 1973; Wu et al., 2004; Roters et al., 2010; Kalidindi, 2015; Karlson et al., 2023) spanning multiple length scales. Numerical methods such as the finite element method (FEM) typically incorporate a physically motivated multi-scale representation of such geometries/features to properly resolve the effects they have across disparate length scales. Common examples of such methods include staggered solution

schemes for coarse- and fine-scale meshes such as the Schwarz alternating method (Mota et al., 2022) or the global–local generalized finite element method (GFEM-gl) (Duarte and Kim, 2008); material formulations that account for mechanical deformation through homogenized properties such as Miehe and Bayreuther (2007) and Geers et al. (2010); variational multiscale (Liu and Marsden, 2018); and reduced order models (Ganapathysubramanian and Zabarar, 2004).

The computational efficiency of multi-scale approaches deteriorates as the complexity of the material model and/or the number of elements increases due to the expense of tasks such as the imposition of plasticity constraints and the numerical integration of auxiliary variables like damage at each quadrature point, which are typically performed at each time/continuation step of a simulation. Consequently, full simulations become untenable as the number of elements increase to the tens to hundreds of millions, even for today's fastest computers (Balzani et al., 2016; Budarapu et al., 2019; Klawonn et al., 2021).

Surrogate models have been developed as an alternative to finite element simulations of high-fidelity models by replacing the most computational expensive routines with approximations that are trained offline using data obtained from previous finite element analyses and/or experiments. Focusing on NN-based approaches, prior work in these types of models include approximations of constitutive and balance laws (Huang et al., 2020; Wu et al., 2020; Aldakheel et al., 2021; Benabou, 2021; Wu and Noels, 2022), homogenization schemes (Wang and Sun, 2018; Wu et al., 2020; Wu and Noels, 2022), and methods that simplify or fully replace finite element analyses (Koeppel et al., 2018, 2019, 2020; Capuano and Rimoli, 2019; Im et al., 2021).

Illustrative examples of surrogate models used to approximate constitutive laws include approaches that predict stresses of history-dependent materials using variants of feed-forward networks (Huang et al., 2020), recurrent neural networks (RNNs) such as Long-Short Term Memory (LSTM) (Benabou, 2021), and the Gated Recurrent Unit (GRU) (Wu et al., 2020; Wu and Noels, 2022). Works such as Wang and Sun (2018), Wu et al. (2020), and Wu and Noels (2022) specifically focus on reducing the computational complexity of traditional multi-scale approaches such as FE<sup>2</sup> by approximating the homogenized fields and properties thus eliminating the need to solve the full boundary-value problem for representative volume elements (RVEs). The authors in Aldakheel et al. (2021) replace the costly numerical integration of an ordinary differential equation (ODE) for a phase-field variable that controls the stress degradation with a NN-based approximation. In Im et al. (2021), the authors construct models that predict the full displacement, stress, or force fields for a given geometry such as a structural frame. In these cases, the surrogate fully replaces the need to run a finite element analysis for the given geometry and set of boundary conditions. In another example, a forcing term is trained to predict the corrections/errors on a coarse mesh for a given boundary-value problem solved on a refined mesh (Baiges et al., 2020). The authors achieved accurate predictions of the deformation for a cantilever beam bending problem using a coarsened mesh and a trained error correction term. In the approaches of Koeppel et al. (2018, 2019, 2020) and Capuano and Rimoli (2019), a machine-learned element is trained to predict a set of fields such as forces for only a single element. The full predictive power of such methods is demonstrated by predicting various quantities of interest in FEM simulations with meshed geometries composed of these machine-learned elements.

The common assumption in the above approaches is that the surrogate model only makes predictions on inputs that are near the data it was trained on. Such limitations are discussed briefly in the concluding remarks of works such as Koeppel et al. (2019), Wu et al. (2020), and Aldakheel et al. (2021), but potential solutions are left as research directions for future work. The degree of uncertainty in these surrogate models' predictions is crucial in determining whether such estimates represent an accurate and physically relevant approximation of the original finite element model. Prior work on uncertainty quantification in machine learning includes predictions of uncertainty due to noise (Yang et al., 2021), and predictions of distributions over possible solutions where training data is sparse or nonexistent (Zhu et al., 2019). Other similar applications are also highlighted in Zou et al. (2022). The uncertainty described in these works pertains to problems where physics-constrained NNs predict fields that satisfy some governing ordinary or partial differential equation subject to prescribed initial and/or boundary conditions. To the authors'

knowledge, uncertainty quantification for hybrid approaches that couple NNs with discretization methods such as FEM for simulating the deformation of solids has not been extensively studied to date.

In the current work, uncertainty estimates of predictions that surrogate models produce are constructed within the framework of a coupled FEM-NN approach similar to those of Koeppel et al. (2018, 2019, 2020) and Capuano and Rimoli (2019). In this approach, the machine-learned elements predict internal forces given displacements for a subregion of a full-scale geometry, while the rest of the meshed geometry contains elements whose internal forces are computed with the traditional evaluation of the stresses via the FEM. Ensembles of Bayesian neural networks (BNNs) (Fort et al., 2019) are used to generate a diverse set of internal force predictions. A novel tolerance interval-inspired approach is introduced to account for uncertainty in the ensemble predictions. Ultimately, this approach is used to assess reliability of the NN predictions for a wide range of possible loading conditions. We believe the proposed method provides a powerful tool for quantifying the uncertainty of predictions for NN-based models when coupled with FEM.

The article is organized as follows: In Section 2, the theoretical background on FEM and NNs is presented. In Section 3, the training and uncertainty estimation procedures are discussed. In Section 4, the proposed procedure is highlighted for a three-element geometry with a single machine-learned surrogate element, and the results are discussed. Concluding remarks are presented in Section 5.

## 2. Theoretical background

### 2.1. Finite element formulation

Consider a spatially discretized domain in the current/deformed configuration at some specified time  $t$  (omitted on domains and fields for brevity) as  $\Omega$  composed of a set of finite elements  $\Omega^e$ ,

$$\Omega_{\text{FEM}} \equiv \bigcup \Omega^e, \tag{1}$$

and a contiguous segment of NN elements  $\Omega_{\text{NN}}^e$ ,

$$\Omega_{\text{NN}} \equiv \bigcup \Omega_{\text{NN}}^e, \tag{2}$$

with an imposed set of boundary conditions

$$\begin{aligned} \mathbf{T} \mathbf{n} &= \bar{\mathbf{t}} \text{ on } \partial\Omega_t \\ \mathbf{u} &= \bar{\mathbf{u}} \text{ on } \partial\Omega_u \end{aligned} \tag{3}$$

as shown in Figure 1. In the above equation,  $\mathbf{T}$  is the Cauchy stress tensor,  $\mathbf{n}$  is the outward-facing unit normal on  $\partial\Omega$ ,  $\mathbf{u}$  is the displacement relative to a stress-free reference configuration, and  $\bar{\mathbf{t}}$  and  $\bar{\mathbf{u}}$  are the imposed set of boundary traction and displacement (respectively). Additionally, the superscript  $e$  signifies a unique element identifier.

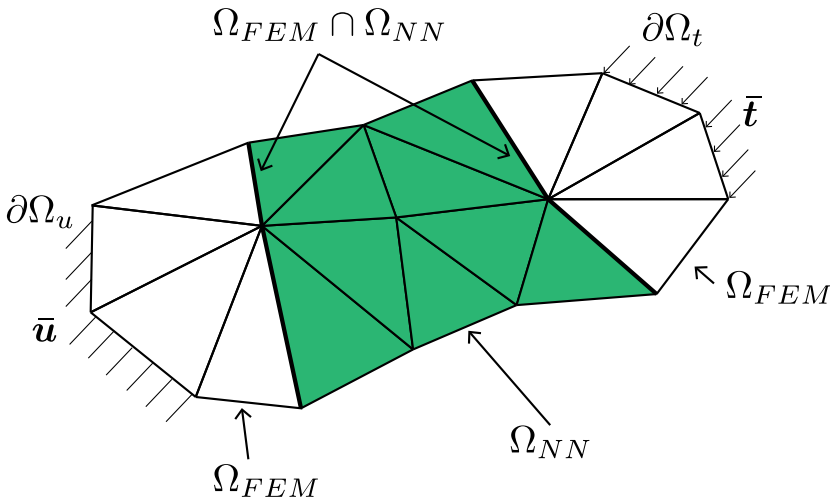
The spatially discretized Bubnov–Galerkin formulation for the balance of static forces for a given finite element  $\Omega^e$  is (Hughes, 2012)

$$[\mathbf{F}_{int}^e(\mathbf{u})] - [\mathbf{F}_{ext}^e] = [\mathbf{0}], \tag{4}$$

where the finite element internal and external forces are defined as

$$\begin{aligned} [\mathbf{F}^{e,int}] &= \int_{\Omega^e} [\mathbf{B}^e]^T \hat{\mathbf{T}} d\mathbf{v} \\ [\mathbf{F}^{e,ext}] &= \int_{\partial\Omega^e \cap \Gamma_t} [\mathbf{N}^e]^T \bar{\mathbf{t}} da + \int_{\partial\Omega^e \setminus \Gamma_t} [\mathbf{N}^e]^T \mathbf{t} da \end{aligned} \tag{5}$$

In the above equation,  $\hat{\mathbf{T}}$  is the Cauchy stress ordered in Voigt notation, and  $[\mathbf{N}^e]$  and  $[\mathbf{B}^e]$  are the traditional arrays of the finite element shape functions and their derivatives.



**Figure 1.** Discretized domain  $\Omega$  highlighting the NN element domain  $\Omega_{NN}$  (green) and finite element domain  $\Omega_{FEM}$ .

Define the linearized strain  $\epsilon$  as the symmetric material gradient of the displacement,

$$\epsilon = \frac{1}{2} \left[ \frac{\partial \mathbf{u}}{\partial \mathbf{X}} + \left( \frac{\partial \mathbf{u}}{\partial \mathbf{X}} \right)^T \right]. \tag{6}$$

Following the classical approach (Simo and Hughes, 1998), we admit an additive decomposition of the strain  $\epsilon$  into an elastic and plastic contribution  $\epsilon^e$  and  $\epsilon^p$  (respectively) at a given point in  $\Omega_{FEM}$  of the form

$$\epsilon = \epsilon^e + \epsilon^p. \tag{7}$$

The (linearized) Cauchy stress  $\sigma$  (here we assume the linearized Cauchy stress  $\sigma$  is equivalent to the Cauchy stress  $\mathbf{T}$ ) is defined as

$$\sigma = \mathbb{C} \cdot \epsilon^e, \tag{8}$$

where  $\mathbb{C}$  is the classical fourth-order elasticity tensor. Plasticity is introduced through inequality constraint on the yield function  $f(\sigma, h)$  which takes the following form,

$$f(\sigma, h) = \phi(\sigma) - (\sigma_y + h(\epsilon^p)) \leq 0. \tag{9}$$

In equation (9),  $\phi(\sigma)$  is the effective stress which, here, is the von Mises criterion,

$$\phi(\sigma) = \sqrt{\frac{2}{3} \mathbf{S} \cdot \mathbf{S}}, \tag{10}$$

where  $\mathbf{S}$  is the deviatoric stress tensor

$$\mathbf{S} = \sigma - \frac{1}{3} \text{tr}(\sigma) \mathbf{I}. \tag{11}$$

Additionally,  $\sigma_y$  is the yield stress, and  $h(\epsilon^p)$  is the isotropic hardening, which is assumed to be a function of the plastic strain  $\epsilon^p$ .

The plastic strain  $\epsilon^p$  is defined assuming associative flow as

$$\epsilon^p = \gamma \frac{\mathbf{S}}{\|\mathbf{S}\|}, \tag{12}$$

where  $\gamma$  is a nonnegative consistency parameter. The Kuhn–Tucker conditions are

$$\begin{cases} f(\boldsymbol{\sigma}, h) \leq 0 \\ \gamma \geq 0 \\ \gamma f(\boldsymbol{\sigma}, h) = 0 \end{cases}, \tag{13}$$

and the consistency condition is

$$\gamma \dot{f}(\boldsymbol{\sigma}, h) = 0. \tag{14}$$

Together, equations (9), (13), and (14) form the full set of plasticity constraints for loading/unloading. The plasticity constraint equations coupled with the constitutive equation in equation (8) provide a closed-form definition of the Cauchy stress needed to evaluate the internal forces in equation (5).

Akin to coupled FEM-NN methods proposed in Capuano and Rimoli (2019) and Koeppel et al. (2020), the internal force term for the NN elements  $\Omega_{NN}^e$  is given by a regression tool such as a NN,

$$[\mathbf{F}_{NN}^{e,int}] \equiv [\mathbf{f}_{NN}(\mathbf{u}^e, \mathbf{w}_{NN}, \mathbf{b}_{NN})], \tag{15}$$

where  $\mathbf{f}_{NN}$  is a nonlinear function of the element displacements  $\mathbf{u}^e$ , NN weights  $\mathbf{w}_{NN}$  and biases  $\mathbf{b}_{NN}$ . The internal forces predicted by  $\mathbf{f}_{NN}$  are, in general, assumed to approximate the relation between the forces and displacements for the material occupying the NN elements. As such, the predicted forces given by  $\mathbf{f}_{NN}$  represent a mechanical response to physically imposed motion, and should thus be trained using a set of relevant forces/displacements.

It is important to mention that the internal force in equation (15) depends on the reference frame used to express the displacements. Additionally, the internal force in equation (15) does not contain any history variables, and thus cannot predict processes that depend on loading history, such as inelastic cyclic loading. Treatment of the reliance of forces/displacements on their respective orientations relative to a fixed reference frame, as well as the dependence of the forces on the history of displacement in the context of FEM-NN approaches, is discussed in Capuano and Rimoli (2019) and Koeppel et al. (2020) (respectively), among other works. In the current context, we assume that the mechanical deformation does not incur large deformations and rotations on elements in the NN domain, and the initial configuration and orientations of the geometry do not change between each training data set and prediction. Additionally, all loads used for training, validation, and prediction are monotonic. These are limitations that simplify the training/prediction of forces in the NN domain. Proper treatment and modifications that account for dependence on reference frame and force/displacement history will be discussed in future work.

The NN element’s balance of forces takes the same form as equation (4), namely,

$$[\mathbf{F}_{NN}^{e,int}(\mathbf{u}^e)] - [\mathbf{F}^{e,ext}] = [\mathbf{0}]. \tag{16}$$

The set of element-wise internal and external forces are assembled across all elements in  $\Omega$  to form the full system of equations,

$$[\mathbf{F}^{int}(\mathbf{u})] - [\mathbf{F}^{ext}] = [\mathbf{0}], \tag{17}$$

where the global internal and external forces are defined as the element-wise assembly of their local counterparts (notated with the assembly operator  $\mathbf{A}$  acting on each element  $e$ ), that is,

$$\begin{aligned} [\mathbf{F}^{int}(\mathbf{u})] &= \mathbf{A}_e \mathbf{F}^{e,int}(\mathbf{u}) \\ [\mathbf{F}^{ext}] &= \mathbf{A}_e \mathbf{F}^{e,ext} \end{aligned} \tag{18}$$

The solution is the set of displacements  $\mathbf{u}$  that satisfies global balance of forces (equation (5)) given the set of boundary conditions in equation (3), a constitutive law governing the relation between the deformation/strains and the Cauchy stress, and an assumed function  $\mathbf{f}_{NN}$  mapping displacements to forces in  $\Omega_{NN}^e$ . The full coupled FEM-NN procedure is shown in Algorithm 1.

The global internal force in equation (18) can be a nonlinear function of the displacements  $\mathbf{u}$  (for either/both the finite element and/or NN element domains). Therefore, obtaining a global displacement vector  $\mathbf{u}$

that satisfies balance of forces across the entire domain  $\Omega$  is typically obtained through iterative methods where each iteration predicts a trial set of displacements/forces to minimize a residual of the form

$$[\mathbf{R}(\mathbf{u})] = [\mathbf{F}^{int}(\mathbf{u})] - [\mathbf{F}^{ext}]. \quad (19)$$

Using numerical integration techniques to evaluate the integral for the internal forces in equation (5) for the traditional finite elements involves the evaluation of the Cauchy stresses at each quadrature point at each iteration. In general, this procedure can be computationally expensive when inelastic deformation takes place due to imposition of plasticity constraints, and in some cases, the evaluation of any state variables that are relevant. In contrast, the evaluation of the NN element internal forces simply involves the computation of  $\mathbf{f}_{NN}$  at each of the element nodes for a given iteration. The lack of physical constraints and numerical solutions of additional variables needed to determine the Cauchy stress can make the evaluation of  $\mathbf{f}_{NN}$  less computationally expensive at the cost of training a NN to predict a set of forces given a set of displacements.

---

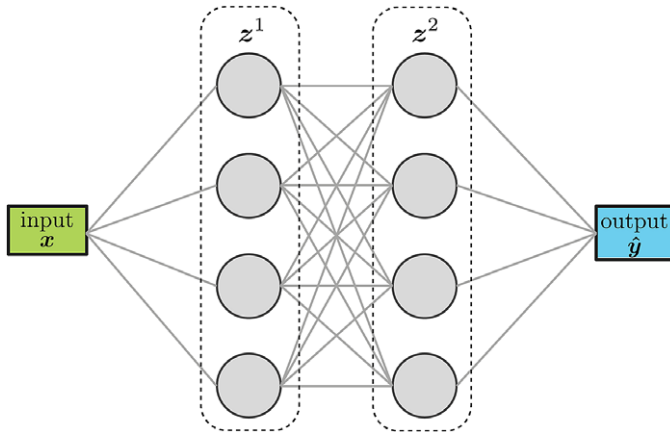
**Algorithm 1.** Coupled FEM-NN procedure

---

- 1: Initialize  $t_0$  and  $\Omega_{t_0}$
  - 2: Set solver parameters  $\text{iter}_{\max}$ ,  $\text{tol}$  ▷ max nonlinear iterations and residual tolerance
  - 3: Set  $i = 1$
  - 4: **while**  $i \leq n_{\text{steps}}$  **do** ▷Begin time loop
  - 5:     Set time step  $\Delta t_i$
  - 6:      $t_i \leftarrow t_{i-1} + \Delta t_i$  ▷Update current time
  - 7:     Initialize trial displacements  $\mathbf{u}_{t_i}^0$
  - 8:     Initialize residual norm  $\|[\mathbf{R}(\mathbf{u}_{t_i}^k)]\|_2$  to large number
  - 9:     Set  $k = 0$
  - 10:     Compute and assemble  $[\mathbf{F}^{e,ext}]$
  - 11:     **while**  $k \leq \text{iter}_{\max}$  and  $\|[\mathbf{R}(\mathbf{u}_{t_i}^k)]\|_2 \geq \text{tol}$  **do** ▷Iterate to obtain equilibrium displacements
  - 12:         Compute and assemble  $[\mathbf{F}_{NN}^{e,int}(\mathbf{u}_{t_i}^k)]$  and  $[\mathbf{F}_{FEM}^{e,int}(\mathbf{u}_{t_i}^k)]$
  - 13:         Update residual  $[\mathbf{R}(\mathbf{u}_{t_i}^k)]$
  - 14:         Compute approximate tangent
  - 15:         Solve for  $\mathbf{u}_{t_i}^{k+1}$
  - 16:          $k \leftarrow k + 1$
  - 17:     **end while**
  - 18:     Extract relevant QOIs
  - 19:      $i \leftarrow i + 1$  ▷Update simulation step
  - 20: **end while**
- 

## 2.2. NN ensembles and Bayesian inference

NNs are a method for approximating nonlinear functional mappings between a set of inputs and a desired set of outputs (LeCun et al., 2015). NN models learn the desired functional mapping from a set of training results and subsequently use the learned function to predict the outputs for new/unseen inputs. NN models are able to establish that functional mapping by identifying an optimal set of features which are then regressed to the desired outputs. The typical transformation NNs use to obtain an output  $z$  from an  $n$ -dimensional input (assumed to be a vector)  $\mathbf{x} \in \mathbb{R}^n$  is called a node (we term this an NN node hereafter to differentiate from nodes of a discretized mesh), and performs the following operation



**Figure 2.** Schematic representation of a feed-forward network with two layers and scalar inputs/outputs.

$$z_i(\mathbf{w}_{\text{NN},i}, \mathbf{x}, b_{\text{NN},i}) = f(\mathbf{w}_{\text{NN},i}^T \mathbf{x} + b_{\text{NN},i}), \quad (20)$$

where  $\mathbf{w}_{\text{NN},i} \in \mathbb{R}^n$  denotes the  $i$ th row of the weights that multiply the input. Similarly,  $b_{\text{NN},i}$  is the  $i$ th bias added to the resultant operation and  $f(\cdot)$  is a nonlinear activation function applied element-wise to its arguments.

Feed forward neural networks (FFNN) sequentially stack layers to transform the inputs into the desired outputs, as illustrated in Figure 2. The number of nodes in a layer as well as the number of layers can be tuned for the specific task at hand. The values of the weights and biases of each layer are obtained by minimizing a loss function that quantifies the discrepancy between the values predicted by the model (*i.e.*, the FFNN) and the known values of the training set. The minimization process is typically stopped when the value of the loss function reaches a pre-determined threshold. FFNNs are a very versatile and accurate surrogate model-building technique that have shown great success in a wide variety of applications and fields (Wang et al., 2018; Wenyoun and Chang, 2020; Zhang and Mohr, 2020; Muhammad et al., 2021).

BNNs have been widely studied for characterizing both aleatoric and epistemic uncertainties (Gal and Ghahramani, 2015; Sargsyan et al., 2015; Kendall and Gal, 2017; Knoblauch et al., 2019). BNNs leverage the concept of Bayesian inference to sample from an assumed posterior distribution given some prior knowledge. Variational Bayesian inference (VBI) (Blundell et al., 2015) is a common approach used to train BNNs. Rather than attempting to sample from the true posterior, which is a computationally intractable problem, VBI assumes a distribution on the weights  $q(\mathbf{w}|\theta)$  where  $\theta$  is the collection of hyperparameters associated with the assumed distribution. The weights are thus sampled from the distribution  $q$ . The hyperparameters  $\theta$  are optimized such that they minimize the variational free energy (also termed expected lower bounds, or ELBO) of the form

$$\mathcal{F}(\mathcal{D}, \theta) = \text{KL}[q(\mathbf{w}|\theta) \| P(\mathbf{w})] - \mathbb{E}_{q(\mathbf{w}|\theta)}[P(\mathcal{D}|\mathbf{w})], \quad (21)$$

where  $P(\mathbf{w})$  is an assumed prior,  $\text{KL}(\cdot)$  is the Kullback–Leibler divergence, and  $\mathcal{D}$  is the training data set. The first term in equation (21) signifies some measure of the distance between the assumed prior  $P(\mathbf{w})$  and the approximate posterior  $q(\mathbf{w}|\theta)$ , which is assumed to always be positive or zero. The last term in equation (21) is a data-dependent likelihood term. The above equation poses an optimization of the approximate posterior  $q$ , with the aim of capturing information about the true posterior based only on knowledge of the assumed prior and the given data. In practice, an approximation of the variational free energy in equation (21) is used to train a BNN. This procedure is explained in Blundell et al. (2015).



The optimization process described above will yield a set of NN parameters that minimize the variational free energy in equation (21). The predictions of an optimized NN generally incur some amount of error. In the context of the FEM-NN approach highlighted in Section 2.1, the replacement of an internal force governed by local stresses that precisely satisfy the constitutive relation in equation (8) and plasticity constraints of equations (9), (13), and (14) with an approximation can commonly lead to a set of global displacements that satisfy the balance of forces in  $\Omega$ . Nevertheless, the approximation can deviate (sometimes significantly) from the true/expected deformation of the body under a given set of boundary conditions. This source of error is epistemic in nature, as it pertains to a lack of knowledge about single-valued (but uncertain) errors from numerous factors such as a finite amount of training data, an assumed form of the NN architecture and loss function, and an optimization of NN weights/biases that is a single realization from an inherently stochastic procedure.

Among the various methods used to quantify and/or control epistemic uncertainty in machine-learning, the concept of NN ensembles (Fort et al., 2019) is particularly relevant for this work. NN ensembles leverage multiple trained NNs termed “base learners.” Common ensembling strategies are randomized-based approaches that concurrently train multiple NNs on the same base data set such as bagging/bootstrapping (Lakshminarayanan et al., 2017), as well as other sequential training methods such as boosting and stacking (Yao et al., 2018). The use of NN ensembles has been shown to substantially improve the quality of predictions, typically by enhancing diversity in those predictions thus reducing overall bias (Lakshminarayanan et al., 2017; Fort et al., 2019; Hüllermeier and Waegeman, 2021). In this work, ensembles are trained concurrently (as is done with traditional bagging approaches). Uncertainty estimates are obtained by bootstrap-sampling from the ensemble multiple times to obtain a distribution of predictions. This procedure is described in Section 3.

### 3. Uncertainty estimation of coupled FEM-NN approaches using ensembles

#### 3.1. Generation of training and validation data

The training and validation data should encompass the sets of input displacements and output forces relevant for a given set of boundary-value problems. In the current context, the inputs and outputs are sampled in  $\Omega_{\text{NN}}$  for a set of relevant boundary conditions. We define these boundary conditions  $\mathcal{B}$  in what is termed the *primary space*,  $\mathbb{P} \subset \mathbb{R}^N$ , where  $N$  corresponds to the dimension of an assumed parameterization of the boundary conditions.

For each boundary condition  $b \in \mathcal{B}$ , there exists a set of “exact” (*i.e.*, not approximated by surrogates) displacements  $\mathbf{u}_b$  that satisfy equilibrium in  $\Omega_{\text{NN}}$ . Typically, these displacements are determined through a traditional finite element simulation. The “exact” internal forces associated with the displacements  $\mathbf{u}_b$  in the elements occupying  $\Omega_{\text{NN}}$  are simply computed using an assumed constitutive law that is a function of the displacements (through its gradient), such as the one shown in equation (8). The combined set of resulting element-level displacements  $\mathbf{D}_{u_b}$  and forces  $\mathbf{D}_{f_b}$  in  $\Omega_{\text{NN}}$  are  $M$ -dimensional vectors defined in the *secondary displacement space* and *secondary force space*,  $\mathbb{S}_u \subset \mathbb{R}^M$  and  $\mathbb{S}_f \subset \mathbb{R}^M$ , respectively. The combined sets of displacements and forces for all boundary conditions in  $\mathcal{B}$  are denoted  $\mathcal{D}_u$  and  $\mathcal{D}_f$  (respectively).

In general, dimensionality reduction via methods such as principal component analysis (PCA) (Jolliffe and Morgan, 1992) can be used to sample relevant boundary conditions in a reduced primary space  $\mathbb{P}_r \subset \mathbb{R}^n$  (where  $n < N$ ), as well as generate training data in a reduced secondary input space  $\mathbb{S}_{ur} \in \mathbb{R}^m$  (where  $m < M$ ). In the current setting, we assume dimensionality reduction is only performed on the secondary input space. We define the linear mapping between secondary and reduced secondary spaces obtained through PCA as  $\varphi_{ur} : \mathbb{S}_u \mapsto \mathbb{S}_{ur}$ . The full input data set of all displacements in its reduced space is denoted  $\mathcal{D}_{ur}$ .

Once the full training and validation set are established, the displacements and forces are each scaled by their standard deviation and normalized to values between  $-1$  and  $1$ , that is,



$$\begin{cases} \hat{D}_{u_b}^i = \frac{\tilde{D}_{u_b}^i}{\max|\tilde{\mathbf{D}}_u|}, \tilde{D}_{u_b}^i = \frac{D_{u_b}^i}{\sigma(\mathbf{D}_u^i)} \\ \hat{D}_{f_b}^i = \frac{\tilde{D}_{f_b}^i}{\max|\tilde{\mathbf{D}}_f|}, \tilde{D}_{f_b}^i = \frac{D_{f_b}^i}{\sigma(\mathbf{D}_f^i)} \end{cases} \quad (22)$$

In the above equation,  $\mathbf{D}_u$  and  $\mathbf{D}_f$  are the full set of displacements  $\mathcal{D}_u$  and forces  $\mathcal{D}_f$  (respectively) in matrix form. Additionally,  $\hat{D}_{u_b}^i$  and  $\hat{D}_{f_b}^i$  are the scaled data for the  $i$ th component/feature of the  $b$ th set of displacements and forces. Lastly,  $\sigma(\mathbf{D}_u^i)$  and  $\sigma(\mathbf{D}_f^i)$  are the standard deviations of the full displacement and force data sets for the  $i$ th component. Note that the standard deviations are computed only for the training data, though the scaling transformations are applied to both training and validation data. The NNs are subsequently trained  $n_e$  times to obtain multiple sets of optimal hyperparameters. The combined set of  $n_e$  NNs forms the ensemble used to make predictions. The full training procedure is shown in Algorithm 2.

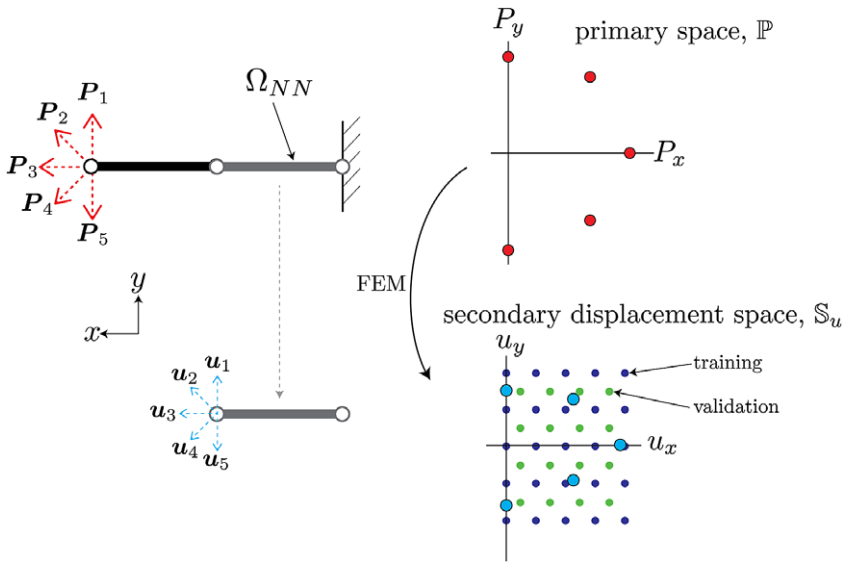
---

**Algorithm 2.** Training procedure

---

- 1: Generate set of relevant boundary conditions  $\mathcal{B} \in \mathbb{P}$
  - 2: **for**  $b \in \mathcal{B}$  **do** ▷ loop through relevant boundary conditions
  - 3:     Compute displacements/forces  $\mathbf{D}_{u_b}$  and  $\mathbf{D}_{f_b}$  in secondary space  $\mathbb{S}$  ▷ via FEM
  - 4: **end for**
  - 5: Compute displacement mapping to reduced secondary space  $\mathbb{S}_{ur}$
  - 6: Construct reduced training/validation input displacement data set  $\mathcal{D}_{ur} \in \mathbb{S}_{ur}$
  - 7: **for**  $b \in \mathcal{B}$  **do**
  - 8:     Compute actual input displacement data  $\mathbf{D}_{u_b} = \phi_{ur}^{-1}(\mathbf{D}_{ur_b})$
  - 9:     Compute output forces  $\mathbf{D}_{f_b}$
  - 10: **end for**
  - 11: split full data sets  $\mathcal{D}_u$  and  $\mathcal{D}_b$  to training/validation  $\{\mathbf{D}_{u,t}, \mathbf{D}_{u,v}\}$  and  $\{\mathbf{D}_{f,t}, \mathbf{D}_{f,v}\}$
  - 12: apply feature scaling to data sets
  - 13: **for**  $n \in \text{range}(n_e)$  **do** ▷ loop through total number of NNs for the ensemble
  - 14:     Randomly initialize weights  $\mathbf{w}_{NN}^n$  and biases  $\mathbf{b}_{NN}^n$  by sampling prior distribution
  - 15:     Find optimal posteriors  $\mathbf{w}_{NN}^n \sim \mathbf{q}_w^n(\theta_w^n)$  and  $\mathbf{b}_{NN}^n \sim \mathbf{q}_b^n(\theta_b^n)$  ▷ Using VBI
  - 16: **end for**
- 

As an illustrative example of the procedure described above, consider a simple two-element cantilever beam embedded in two-dimensional space  $\mathbb{R}^2$ , as shown in Figure 3. The left end that has an applied load can be parameterized by variations in both the horizontal and vertical directions, therefore  $N = 2$ . The set  $\mathcal{B}$  consists of variations of the applied load in the parameterized two-dimensional space, shown as the red points on the top plot in Figure 3. The NN-element has two independent degrees of freedom for its displacement and force on the left node, therefore  $M = 2$ . The element-level displacements  $\mathbf{D}_{u_b}$  corresponding to the applied loads in primary space are shown as blue circles in the secondary space plot in Figure 3. Note that the force on the right node of the NN-element is always equal and opposite to that of the left node, and thus does not have independent degrees of freedom. For simplicity, the training and validation input data can be constructed as a grid of displacements in the reduced secondary space such that they span the original data  $\mathbf{D}_{u_b}$ , as schematically shown in Figure 3. In this setting, an interpolative prediction occurs when the input displacements lie within the grid, whereas extrapolative predictions encompass all the input space not spanned by the grid. For this problem, dimensionality reduction is not strictly needed since both primary and secondary spaces are two-dimensional.



**Figure 3.** Schematic of sampled boundary conditions in primary space  $\mathbb{P}$  (top) and the corresponding displacements used to inform training/validation in secondary space  $\mathbb{S}$  (bottom). The circles attached to either side of the beam are the nodes associated with the degrees of freedom of this element. The red and blue arrows on the left diagrams correspond to the red and blue points on the right plots.

### 3.2. Uncertainty estimation procedure

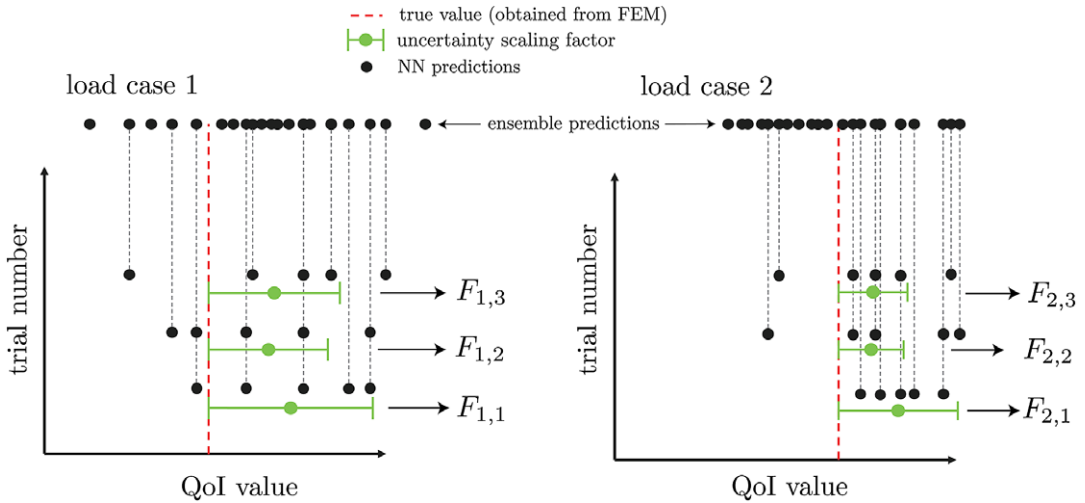
For a given ensemble  $m$  of a load case  $T \in \mathcal{T}$  (with index  $\tau$ ), we define a statistical tolerance interval (Hahn and Meeker, 1991) inspired uncertainty scaling factor  $F_{\tau,m}$  of the ensemble’s standard deviation  $\sigma_{\tau,m}$  such that the true value of the QoI (obtained via FEM) denoted  $f_{\tau,m}^{\text{FEM}}$  is  $\pm F_{\tau,m}\sigma_{\tau,m}$  from the predicted mean  $\mu_{\tau,m}$ . With this requirement, the uncertainty scaling factor  $F$  can be expressed as

$$F_{\tau,m} = \frac{e_{\tau,m}}{\sigma_{\tau,m}}, \tag{23}$$

where the mean bias (absolute error of the mean prediction) is defined as  $e_{\tau,m} = |f_{\tau,m}^{\text{FEM}} - \mu_{\tau,m}|$ . Based on the form of equation (23),  $F$  can be interpreted as a metric of bias-to-variance ratio for a set of NN predictions. High values of  $F$  indicate that the subensemble predictions have a high mean bias and/or low standard deviation (and variance), whereas lower values of  $F$  indicate a low mean bias and/or high standard deviation.

To account for the load-dependent variation in the uncertainty scaling factor  $F$ , a selected set of test/trial boundary conditions  $\mathcal{T}$  are constructed and sampled in primary space  $\mathbb{P}$ . For each boundary condition instance, the full boundary-value problem is solved via the FEM-NN method highlighted in Algorithm 1, and the predictions for each quantity of interest (QoI) are extracted at equilibrium. In this prediction process, each NN of the ensemble produces a different output due to the stochastic nature of training these networks via randomly initialized weights/biases, back-propagation (e.g., using stochastic gradient descent), and the computation of an approximate variational free energy obtained via random sampling. The NN ensemble will thus produce a set of  $n_e$  predictions for a given test/trial case.

In practice, producing an ensemble-averaged prediction can be prohibitively expensive, as it requires the evaluation of  $n_e$  NNs for a given load/boundary condition. A subensemble of  $n_{ss}$  BNNs can be selected via random sampling to reduce the computational cost of the ensemble-averaged predictions. The number of NNs  $n_{ss}$  that the subensembles contain is typically selected to optimize the cost of making predictions in a typical application setting. For instance, large models involving complicated discretized geometries may require smaller subensembles to maintain reasonable computational efficiency.



**Figure 4.** Extraction of uncertainty for three sampled neural network subensembles for two sampled load cases with  $n_e = 20$  and  $n_{ss} = 5$ .

To estimate uncertainty due to variations in subensemble predictions, we generate  $n_{se}$  unique combinations of subensembles, each composed of  $n_{ss}$  BNNs. Each prediction of a given load case will have a scaling factor  $F$  based on its mean bias and standard deviation, as illustrated in Figure 4. With this set of information, each test case will have a distribution of  $F$  values, with the lowest ones representing the subset of NNs that are most optimal (*i.e.*, lowest mean bias and/or highest standard deviation). The uncertainty estimation calibration procedure described here is highlighted in Algorithm 3. Note that the prediction of a given BNN statistically varies, due to the uncertain nature of sampling from an approximate posterior. Therefore, this scaling factor  $F$  accounts both for uncertainty between each BNN prediction as well as the uncertainty within a single BNN.

---

**Algorithm 3.** Uncertainty estimation procedure

---

- 1: Generate set of test loading conditions  $\mathcal{T} \in \mathbb{P}$
  - 2:  $\tau \leftarrow 0$  ▷ Initialize test boundary condition counter
  - 3: **for**  $T \in \mathcal{T}$  **do**
  - 4:     Compute “exact” QoIs for BCs associated with  $T$  ▷ via high-fidelity FEM simulation
  - 5:     **for**  $n \in \text{range}(n_e)$  **do**
  - 6:         Run FEM-NN simulations using  $\mathbf{w}_{NN}^n$  and  $\mathbf{b}_{NN}^n$  to compute internal forces  $F_{int,NN}^e$
  - 7:         **if** run succeeds **then**
  - 8:             Save QoIs
  - 9:         **end if**
  - 10:     **end if**
  - 11:     **for**  $m \in \text{range}(n_{se})$  **do**
  - 12:         Randomly select  $n_{ss}$  subsamples of QoIs from successful runs
  - 13:         Compute mean bias  $e_{\tau,m}$  and standard deviation  $\sigma_{\tau,m}$  of QoIs
  - 14:         Compute ensemble’s uncertainty scaling factor  $F_{\tau,m} = e_{\tau,m} / \sigma_{\tau,m}$
  - 15:     **end for**
  - 16:      $\tau \leftarrow \tau + 1$  ▷ Increment test boundary condition counter
  - 17: **end for**
-

### 3.3. Calibration and reliability of uncertainty estimates

In the current setting, we assume  $n_e = 20$  is a representative number of total NNs that is sufficient to capture reasonable diversity in predictions. Moreover,  $n_{ss} = 5$  is chosen as a reasonable size for a subensemble that is relatively affordable for our applications. Lastly, the number of subensembles where we compute an uncertainty scaling factor is  $n_{se} = 100$ . Therefore, for each test point  $T \in \mathcal{T}$ , there exist 100 values of  $F$ , corresponding to the 100 different combinations of ensemble prediction biases/variances. The subensembles are randomly chosen (with a fixed random seed for reproducibility) from the  $n_e = 20$  ensembles such that each combination is unique.

The uppertail of the distribution of 100  $F$  values is a useful metric of the predictive uncertainty for this problem, since the highest values of  $F$  indicate the least reliable predictions (*i.e.*, highest bias and/or lowest variance) generated by the subensembles for a given load. More specifically, we assume that the 95<sup>th</sup>-percentile of  $F$  (for short will be notated  $F^{95}$ ) for a given test point  $T$  is sufficient to provide conservative estimates of the higher-end of bias/variance seen in the predictions while excluding outliers that detract from the primary trends. Note that this selection of an upper-bound is a heuristic parameter that is problem-dependent, and should be fine-tuned based on the observed distribution of  $F$  values.

Since  $F$  varies spatially across the primary space  $\mathbb{P}$ , we introduce a continuous polynomial uncertainty estimator function  $\mathcal{F}$  whose coefficients are calibrated to match the  $F^{95}$  values at points in the calibration set  $\mathcal{T}_{\text{calibration}} \in \mathbb{P}$ . The uncertainty scale factor  $F$  and its 95<sup>th</sup> percentile (notated  $F^{95}$ ) are known at each point of the calibration set. Formally, the uncertainty estimator maps any point  $p \in \mathbb{P}$  to some interpolated value  $F_p^{95}$ , that is,  $\mathcal{F}: p \in \mathbb{P} \mapsto F_p^{95}$ . The subscript  $p$  indicates the function  $\mathcal{F}$  is evaluated at point  $p$ . For simplicity, linear interpolation of  $F^{95}$  values between calibration points in  $\mathcal{T}_{\text{calibration}}$  is used. Additionally, a point  $r$  outside of  $\mathbb{P}$  assumes a value of  $F^{95}$  equivalent to the estimator function  $F$  evaluated at the point on the boundary of  $\mathbb{P}$  closest to  $r$ .

The reliability of the estimator  $\mathcal{F}$  is determined by its ability to predict conservative  $F^{95}$  values at other points between or outside of the calibration set, termed the *test set*  $\hat{\mathcal{T}}$ . We define  $P^{95}$  as the probability that the 95<sup>th</sup>-percentile of computed  $F$  values at a given test point  $\hat{T}$  (labeled  $\hat{F}_{\hat{T}}^{95}$ ) is less than or equal to the estimated  $F^{95}$  value obtained from evaluating  $\mathcal{F}$  at the point  $\hat{T}$  (labeled  $F_{\hat{T}}^{95}$ ), that is,

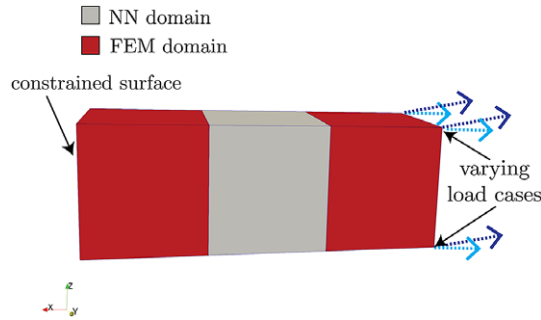
$$P^{95} = p(\hat{F}_{\hat{T}}^{95} \leq F_{\hat{T}}^{95}). \quad (24)$$

The value of  $P^{95}$  can also be interpreted as the probability that the true QoI value obtained from FEM lies within the range  $\pm F_{\hat{T}}^{95} \sigma_{\hat{T},m}$  from the mean  $\mu_{\hat{T},m}$  for a given subensemble  $m$  and a given test case  $\hat{T}$ . For instance, with  $n_{se} = 100$  and  $P^{95} = 0.90$  for some set of boundary conditions, the mean predictions of 90 of the 100NN subensembles will lie within at most  $\pm F_{\hat{T}}^{95} \sigma_{\hat{T},m}$  of the true value. In other words, the estimated uncertainty scale factor at this boundary condition  $\hat{T}$  will lead to a 90% empirical success rate at capturing the true value.

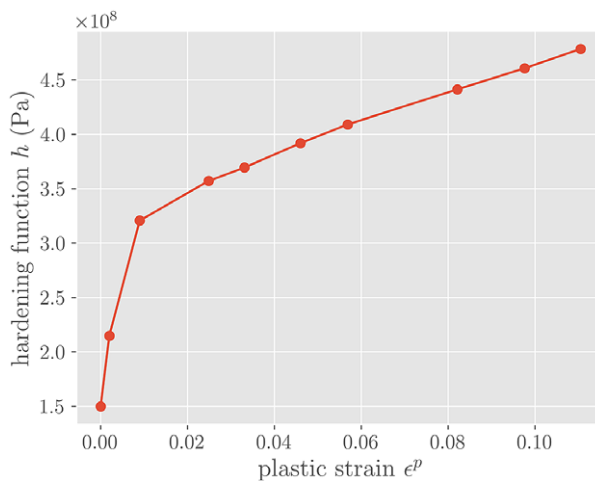
## 4. FE-NN-UQ performance characterization test problem

### 4.1. Results

Here, we demonstrate the procedure and implementation described in Section 3 on a simple FEM model that consists of a rectangular bar 3 cm long and 1 cm wide, meshed into three elements of equal volume, as shown in Figure 5. This bar is fully constrained in the  $x$ -,  $y$ -, and  $z$ -directions (no translations/rotations) on its left-most surface along the  $x$ -axis, and is exposed to some varying imposed displacements (in both direction and magnitude) on its right-most surface. The bar is composed of a generic type of steel, with a Young's modulus, Poisson's ratio, and yield stress of 200 GPa, 0.27, and 150 MPa, respectively.  $J_2$ -plasticity of the form shown in equation (10) is assumed for the middle element in Figure 5, with an isotropic hardening term  $h(\epsilon^p)$  that is the piece-wise linear function shown in Figure 6. The left-most and right-most elements are linearly elastic with the aforementioned Young's modulus and Poisson's



**Figure 5.** Geometry and boundary conditions of three-element model.



**Figure 6.** Hardening function  $h(\epsilon^p)$ .

ratio. Each of the three elements is an eight-noded linear hexahedron, and assumes two Gauss–Legendre integration points per direction. The QoI used to evaluate the uncertainty is the integration-point-averaged von Mises stress on the right-most element in Figure 5. All FEM and coupled FEM-NN simulations are conducted in SIERRA/SM, a Lagrangian finite element code for solids and structures (SIERRA Solid Mechanics Team, 2021).

In practice, the region where the NN-elements are placed should coincide with portions of the mesh that are either spatially varying (*i.e.*, having geometric defects such as notches or voids), or consist of material that undergoes inelastic deformation. The former requires either a homogenization scheme or proper mesh refinement of these features. This is discussed to some extent within the context FEM-NN methods in Capuano and Rimoli (2019). Here, we focus on the latter, by replacing the middle FEM-based element that has an inelastic material response with a NN-element.

Training data is obtained by extracting the forces/displacements on the nodes connected to the NN-element through a series of FEM simulations. For training, both FEM and NN domains compute the internal force directly through the volume integration of the stress divergence term, that is, equation (5). Each FEM simulation has applied boundary displacements sampled from the primary set space  $\mathbb{P} \subset \mathbb{R}^3$  spanning  $[-4, -2] \times [-1, 1] \times [-1, 1]$  (units: mm), where the first, second, and third dimensions correspond to displacements along the  $x$ -,  $y$ -, and  $z$ -directions (per the axes shown in Figure 5), respectively. Here, it is assumed that the applied displacement on all four nodes of the boundary is equal in magnitude and direction. Therefore, a single point in  $\mathbb{P}$  corresponds to a 3-dimensional displacement

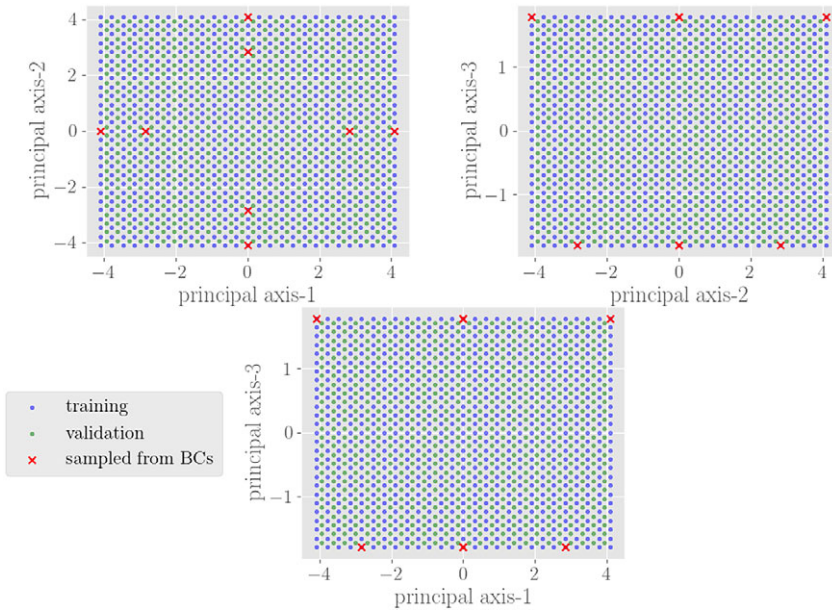
vector that is assigned to each of these four nodes. The volume-averaged plastic strain for boundary conditions within the selected region  $\mathbb{P}$  range from approximately 14–29%.

Boundary conditions are sampled in  $\mathbb{P}$  to both inform the relevant domain of input space used for training, and serve as calibration points for the uncertainty estimator. In this setting, the set  $\mathcal{B}$  is identical to  $\mathcal{T}_{\text{calibration}}$ . To start, this training/calibration set consists of the eight corners of  $\mathbb{P}$ , which is a  $2 \times 2 \times 2$  grid denoted as  $\mathcal{T}_{2 \times 2 \times 2}$ . The nodal displacements and internal forces of the middle element obtained from the eight FEM simulations are the boundary of the training/validation secondary data space  $\mathcal{S}$ . Since there are eight nodes in this element and each node has three degrees of freedom, the input (displacement) and output (force) spaces are both 24-dimensional, that is,  $\mathcal{S} \subset \mathbb{R}^{24}$ .

Obtaining training data for every possible combination of displacements/forces the NN can predict is intractable due to the large number of samples required. As an alternative, training and validation displacements/forces are sampled and computed on a discretized set spanning  $\mathcal{S}$ . For simplicity, discretization of the training/validation input space is performed by creating a cube grid with equidistant intervals between each point in a reduced secondary space  $\mathcal{S}_{ur} \subset \mathbb{R}^3$ . The mapping  $\varphi_{ur} : \mathcal{S}_u \mapsto \mathcal{S}_{ur}$  is obtained via PCA. In this case, the reduced input displacement data in three-dimensions has a total explained variance of 98% of the original 24-dimensional data. The sampled training/validation input points are shown in Figure 7. The number of training and validation grid points per direction is 27 and 26 (respectively), leading to approximately 20k and 17.5k total data points (respectively). To obtain the displacement data set  $\mathcal{D}_{ur}$  in the reduced space  $\mathcal{S}_{ur}$ , the nodal forces are computed through a single-step FEM simulation of the NN-element with each of the 24 degrees of freedom constrained by the displacements corresponding to this boundary condition. The 24 components of the displacement can be obtained by applying the inverse mapping on the input data in the reduced space, which for a given set of boundary conditions  $b \in \mathcal{B}$  is expressed as

$$\mathbf{D}_{u_b} = \varphi_{ur}^{-1}(\mathbf{D}_{ur_b}). \tag{25}$$

The NN architecture consists of an input layer with 24 NN nodes for the nodal displacements, a 100 NN node hidden layer with a sigmoid activation function, and a 24 NN node output layer for the forces. The



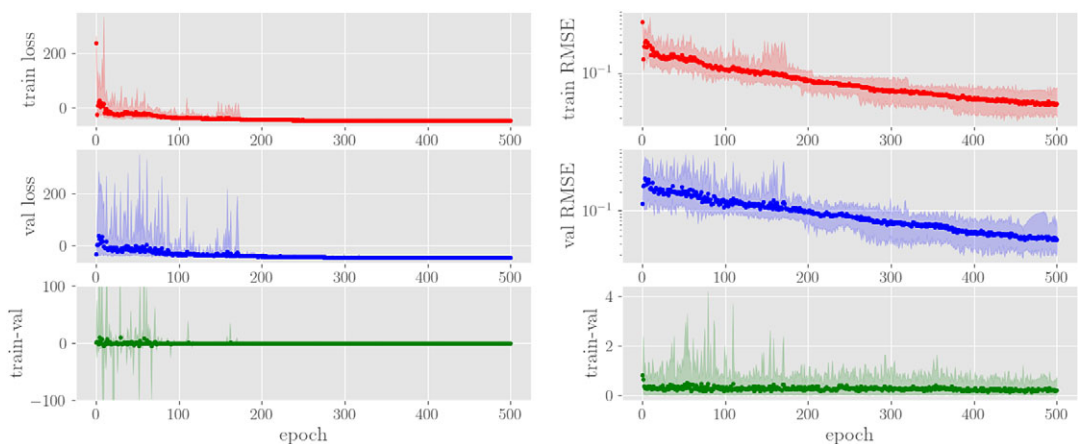
**Figure 7.** Grid of training and validation data shown in cross sections of the 1–2 (top-left), 2–3 (top-right), and 1–3 (bottom-center) principal axes in reduced secondary space  $\mathcal{S}_{ur}$ . Red data points correspond to displacements associated with each of the eight corner points in  $\mathbb{P}$ . The densely packed blue and green grid points represent the training and validation data, respectively. Values on all axes are scaled by  $1.0e3$ .



choice of a single hidden layer with 100 NN nodes is a heuristic parameter that was determined to provide sufficient interpolation accuracy while avoiding overfitting (Montes de Oca Zapiain et al., 2022). Additionally, the use of a sigmoid activation function for the hidden layer allows for a nonlinear mapping between input displacements and output forces that has smooth gradients. The assumed prior for all the NN weights is a normal distribution with mean of 0 and standard deviation of  $e^{-2}$ . The weight means and standard deviations are randomly initialized from a normal distribution with 0 mean and a standard deviation of 0.1, and the bias means and standard deviations are randomly initialized from a uniform distribution in the closed range  $[0, 4]$ . These prior distributions and initializations of the weights and biases were selected through trial-and-error, with the goal of balancing accuracy and prediction diversity.

The training of each BNN is performed on a single CPU using the PyTorch library (Paszke et al., 2019). The weight/bias means and standard deviations are chosen to optimize the variational free energy, that is, equation (21). The training takes place for 500 epochs, using the Adam optimizer (Kingma and Ba, 2014). The initial learning rate is 0.1, and is halved when the loss remains stagnant to within a  $1.0e-4$  tolerance for more than 50 epochs. The total number of epochs and learning rate used for training were selected through trial-and-error. In general, selecting more epochs did not improve the prediction accuracy, and training for less epochs led to a large variability between each NN prediction such that their overall accuracy was diminished. As highlighted in Figure 8, the total variational free energy and root-mean-square error (RMSE) of both training and validation data decrease proportionally for all 20 NNs of the full ensemble, despite the initially large network-to-network variability, particularly during the first 200 epochs. Note that the variational free energy stabilizes at a negative value, which in this setting, indicates that the likelihood cost term in equation (21) is greater than the KL divergence term. The RMSE of both the training and validation data stabilizes at values ranging from  $2.0e-2$  to  $6.0e-2$ , with a mean near  $3.0e-2$ . Additionally, the validation RMSE deviates by an average of 8% from the training RMSE. With these combined observations, we claim that the trained BNNs are sufficiently accurate in making predictions at training points, and are not overfit when predicting at the validation points.

The uncertainty estimates of the NN ensembles are obtained according to Algorithm 3 on the calibration set  $\mathcal{T}_{2 \times 2 \times 2}$  that was used to perform the training. Estimators are also established for refined discretizations of the primary space  $\mathbb{P}$ , namely for the sets  $\mathcal{T}_{3 \times 3 \times 3}$ ,  $\mathcal{T}_{4 \times 4 \times 4}$ , and  $\mathcal{T}_{5 \times 5 \times 5}$ , in addition to the original set  $\mathcal{T}_{2 \times 2 \times 2}$ , as shown in Figure 9. For short, these collective sets and the corresponding uncertainty estimators will be notated  $\mathcal{T}_{i \times i \times i}$  and  $\mathcal{F}_{i \times i \times i}$  (respectively), with the assumption that  $i = 2 \dots 5$ .



**Figure 8.** Average training loss (red), validation loss (blue), and relative difference (green) of all 20 NNs based on approximate variational free energy shown in equation (21) (left three plots) and RMSE (right three plots, log-scaled). Averages are shown as dots. Shading corresponds to range of values across 20 networks. Outliers of validation deviations for initial epochs are not shown for clarity.



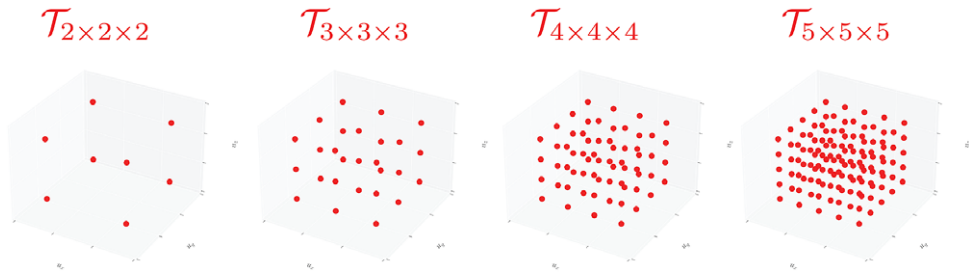


Figure 9. Grid of primary space sets  $\mathcal{T}_{2 \times 2 \times 2}$ ,  $\mathcal{T}_{3 \times 3 \times 3}$ ,  $\mathcal{T}_{4 \times 4 \times 4}$ , and  $\mathcal{T}_{5 \times 5 \times 5}$ .

To determine whether the training data informed by the boundary conditions in  $\mathcal{T}_{2 \times 2 \times 2}$  covers the range of inputs for each of the refined test cases ( $i = 3, 4, 5$ ), FEM simulations are conducted for each boundary condition to obtain the set of inputs (nodal displacements)  $\mathcal{D}_{u,i \times i \times i}$  on the NN-element. These displacements are subsequently mapped to the reduced training space using the transformation  $\varphi_{sr}$  to determine whether any of the reduced set of displacements  $\mathcal{D}_{ur,i \times i \times i}$  lie outside of the original training bounds obtained from the boundary conditions of  $\mathcal{T}_{2 \times 2 \times 2}$ . As shown in Figure 10, the original training input data set spans inputs of all refined cases. Therefore, the same set of NNs can be used to make predictions for all test cases in consideration without a need for retraining.

The test set  $\hat{\mathcal{T}}$  contains points sampled from the test primary space  $\hat{\mathbb{P}}$ , which both fully encloses and contains regions outside of the original primary space  $\mathbb{P}$ , that is,  $\mathbb{P} \subset \hat{\mathbb{P}}$ . The region  $\hat{\mathbb{P}}$  is the enclosed three-dimensional space  $[-4.5, -1.5] \times [-1.5, 1.5] \times [-1.5, 1.5]$  (units: mm). We assume a uniformly distributed  $8 \times 8 \times 8$  grid defines the set of test points  $\hat{\mathcal{T}}_{8 \times 8 \times 8} \in \hat{\mathbb{P}}$ . The subset of test points that lie inside

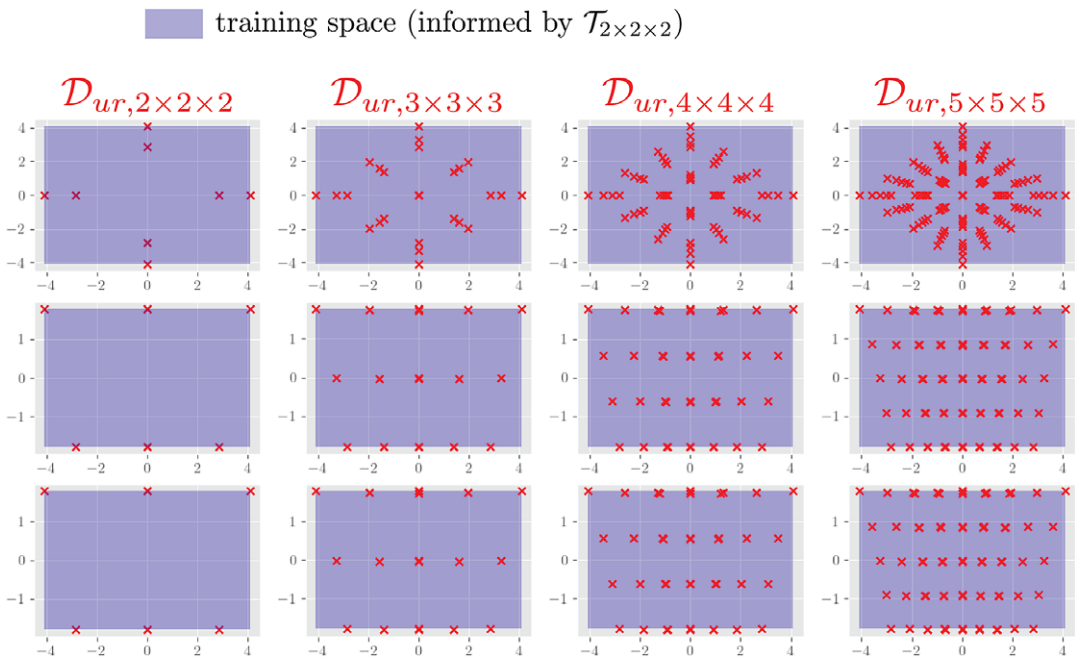
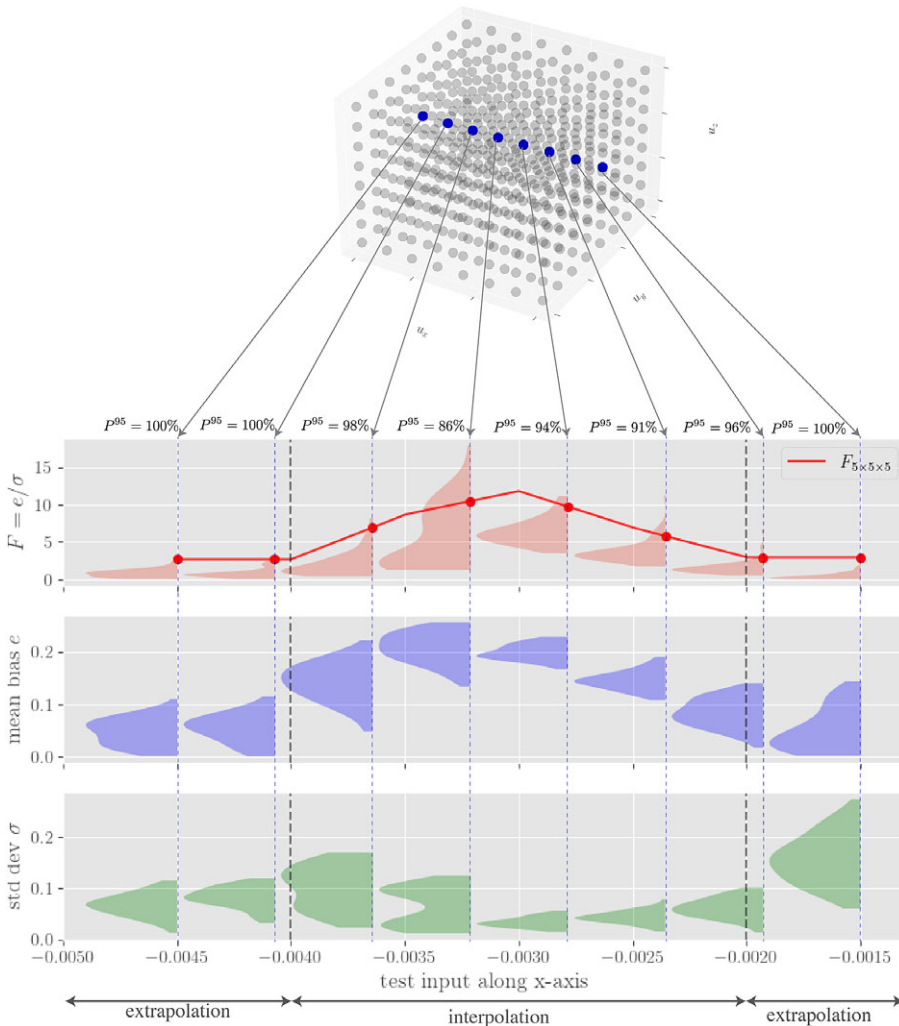


Figure 10. Values of displacements on the NN-element in the reduced space  $\mathcal{S}_{ur}$  (red markers) associated with all refined discretizations of boundary conditions in the primary space  $\mathcal{T}_{i \times i \times i}$  for  $i = 2 \dots 5$ . Cross-sections are shown for the principal 1–2 (top), 2–3 (middle), and 1–3 axes (bottom). The space of input data used for training/validation is shaded in purple. Values on all axes are scaled by  $1e3$ .

(interpolative predictions) and outside (extrapolative predictions) of  $\mathbb{P}$  are notated  $\hat{T}_{8 \times 8 \times 8}^{\text{interp}}$  and  $\hat{T}_{8 \times 8 \times 8}^{\text{extrap}}$ , respectively. FEM-NN simulations are run for each test point to generate predictions for each of the 20 NNs. The same procedure used for the calibration set is repeated for the test set, with the identical values  $n_e = 20$ ,  $n_{ss} = 5$ , and  $n_{se} = 100$ . Solutions that do not converge are discarded in analyzing uncertainty, which comprise approximately 5.5% and 6.8% of the total prediction attempts for interpolation and extrapolation (respectively). If, for instance, 19 of the 20 total NN predictions for a specific load case result in globally converged residuals, each combination of five NN subensembles are randomly selected from the 19 converged predictions rather than the full set of 20.

To illustrate the evaluation of an uncertainty estimator with respect to an observed set of uncertainty scale factors, consider the set of eight test points highlighted in blue in Figure 11. The distributions of  $\hat{F}^{95}$  values from the 100 subensemble predictions are compared against the uncertainty estimator  $\mathcal{F}_{5 \times 5 \times 5}$  on the top plot shown in Figure 11. The distributions of  $\hat{F}^{95}$  clearly vary by test case, with some of the highest

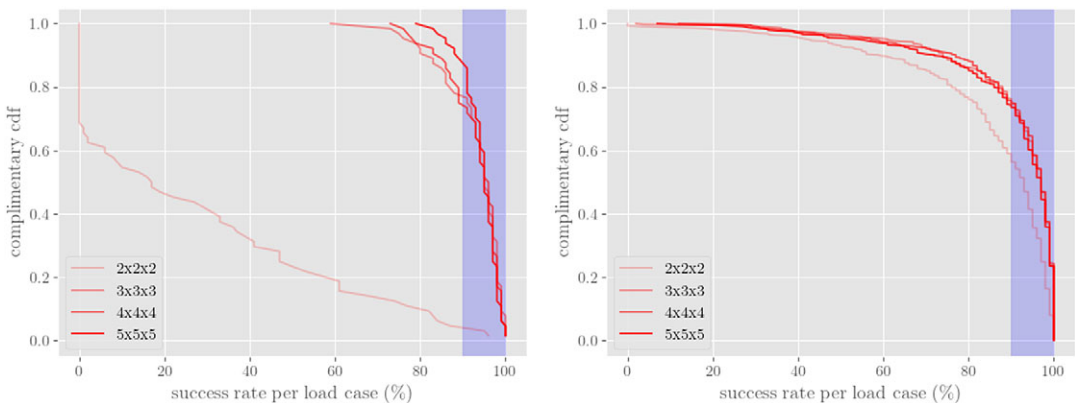


**Figure 11.** Distributions of the observed uncertainty scale factors  $\hat{F}^{95}$  (top), mean bias  $e$ , and standard deviation  $\sigma$  of the blue test points in  $\hat{T}_{8 \times 8 \times 8}$ . The red curve on the top plot is the uncertainty estimator  $\mathcal{F}_{5 \times 5 \times 5}$  with the red markers identifying the points in the test set where the estimator is evaluated.

values occurring near the middle along the  $x$ -axis. These points tend to exhibit high mean bias and low standard deviation, thus leading to high  $\hat{F}^{95}$  values, as highlighted by the distributions in the middle and bottom plots of Figure 11. This indicates that the predictions will not vary significantly between each NN in an ensemble, but they may generally be biased toward certain predictions that may be inaccurate. Interestingly, predictions of the four extrapolation test points show low mean bias and low standard deviation (leading to low  $\hat{F}^{95}$  values), meaning that the NN ensembles produce a wide set of predictions, some of which may have lower errors.

The reliability of the uncertainty estimator is determined by the proportion of total test cases in some region of the primary test space  $\hat{\mathbb{P}}$  that have a specified minimum acceptable success rate per load case, which we denote  $P_{acc}^{95}$ . For instance, suppose we assume that a value of  $P_{acc}^{95} = 90\%$  is sufficiently conservative in predicting the 95<sup>th</sup> percentile of observed uncertainty scale factors  $\hat{F}^{95}$ . For the population of eight test points shown in Figure 11, 7/8 or 87.5% of the considered test cases have a success rate per load case of at least 90%. In other words, the estimator  $\mathcal{F}_{5 \times 5 \times 5}$  will predict an uncertainty scaling factor  $F^{95}$  that has at least a 90% probability of capturing the true QoI value for 87.5% of the eight considered test cases. Note that the minimum success rate per load is generally problem-dependent and should be decided based on the application.

To evaluate the full set of test data in a similar fashion as described above, we determine the distributions of  $\hat{F}^{95}$  for each point in  $\tilde{T}_{8 \times 8 \times 8}$ , and compute the corresponding  $P^{95}$  value using each uncertainty estimator  $\mathcal{F}_{i \times i \times i}$ . The empirical complimentary cumulative distribution functions (CCDFs) of all  $P^{95}$  values using each of the four calibrated estimators are shown in Figure 12. Here, we assume  $P_{acc}^{95} = 90\%$ , therefore the coverage (shown as the shaded blue region in Figure 12) spans the closed range [90%, 100%]. It is clear from this figure that the estimator  $\mathcal{F}_{2 \times 2 \times 2}$  significantly underpredicts the 95<sup>th</sup>-percentile upper bounds of test cases, particularly for points where predictions are interpolated within the primary space  $\mathbb{P}$ . In contrast, the estimators of each of the refined cases are significantly more reliable, with approximately 70–85% of the total test cases lying within the desired coverage. This contrast between the CCDF of all refined cases and the CCDF using the estimator calibrated from  $T_{2 \times 2 \times 2}$  indicates that NN ensemble predictions on the corners of the training space  $\mathbb{P}$  may differ significantly from those inside the domain of  $\mathbb{P}$ , and that accounting for these differences and associated uncertainties in predictions is crucial in determining the reliability of the uncertainty estimator.



**Figure 12.** Empirical complimentary CDF functions of  $P^{95}$  values for test cases corresponding to interpolation (left) and extrapolation (right) (relative to the primary space  $\mathbb{P}$ ) using the estimators  $\mathcal{F}_{2 \times 2 \times 2}$ ,  $\mathcal{F}_{3 \times 3 \times 3}$ ,  $\mathcal{F}_{4 \times 4 \times 4}$ , and  $\mathcal{F}_{5 \times 5 \times 5}$ . The blue region indicates the acceptable coverage for the success rates.

#### 4.2. Summary and motivation for future work

With the example in Section 4.1, we highlight the proposed framework of estimating uncertainty on a simple three-element problem undergoing inelastic deformation. To start, a set of boundary conditions used to inform the NN training were parameterized in the primary space  $\mathbb{P}$ , which here is a cube region in  $\mathbb{R}^3$ , where each axis corresponds to a direction of an imposed displacement. The training/validation input displacements were obtained in a reduced three-dimensional secondary space  $\mathbb{S}_{ur}$  from a simple grid constructed around the reduced NN-element displacements at equilibrium for the selected boundary conditions. The training output forces were subsequently determined through FEM. A set of 20 BNNs were trained simultaneously on the same set of input displacement and output force data to optimize the variational free energy. The training/validation plots in Figure 8 highlight that the 20 NNs initially exhibited stable and similar variational free energies and RMSE losses after approximately 200 epochs for the training and validation data, which indicate the BNNs are reasonably accurate and are not overfit when interpolating between training data.

The trained BNNs were used to approximate the internal forces of the middle element in Figure 5 in a series of coupled FEM-NN simulations. The predictions of 100 unique random samples of 5 BNNs from the full set of 20 networks were collected for every relevant boundary condition used for calibration and testing. Each subensemble is viewed as a possible realization of a sparse set of networks used to predict the internal forces in an actual application. The uncertainty scaling factor  $F$ , defined as a multiplier on the prediction set's standard deviation such that the true value of a given QoI is  $\pm F\sigma$  from the predicted mean, is computed for each subensemble.

Every set of 100 uncertainty scaling factors associated with some boundary conditions has its own empirical distribution. This is illustrated for eight representative test boundary conditions in the top plot of Figure 11. Estimates of a sufficiently conservative upper bound for these distributions provides a basis for determining a conservative range of possible predictions for any random ensemble of five BNNs in an application prediction setting. Here, we chose the 95<sup>th</sup>-percentile of the distribution of  $F$  values (termed  $F^{95}$ ) as a sufficiently conservative metric for inferring an ensemble's possible prediction error and consequent uncertainty.

The uncertainty scaling factors and their distributions depend on the applied boundary conditions, and hence vary at any given point in the space  $\mathbb{R}^3$  defining the primary space  $\mathbb{P}$ . For the three-element problem in Section 4.1, the pertinent boundary conditions for predictions consists of a cube region in  $\mathbb{R}^3$  defined as the test primary space  $\hat{\mathbb{P}}$ , which covers the original primary space  $\mathbb{P}$  as well as regions outside of this space. This test primary space was discretized into an  $8 \times 8 \times 8$  grid, with the assumption that this level of refinement is both computationally affordable for this specific setting and is sufficient in capturing the variations in the distributions of the uncertainty scaling factors with varying boundary conditions.

The estimates of  $F^{95}$  values within  $\mathbb{P}$  were computed via a set of piecewise linear functions spanning this space. The coefficients of these functions were fit to match the  $F^{95}$  value for a specific set of coarse grid points spanning the original primary space  $\mathbb{P}$ . These grid points, termed the calibration set, were discretized with multiple levels of refinement, as grids of  $2 \times 2 \times 2$ ,  $3 \times 3 \times 3$ ,  $4 \times 4 \times 4$ , and  $5 \times 5 \times 5$  points, shown in Figure 10. An  $F^{95}$  piecewise function was constructed for each of the four levels of grid refinements. Empirical success rates of uncertainty estimates based on interpolated/extrapolated  $F^{95}$  values were determined by computing the likelihood that these estimates captured the true value of a QoI obtained from FEM at the observed  $8 \times 8 \times 8$  test points. The CCDFs of the full set of success rates for each of the four grid refinements are shown in Figure 12, and are qualitatively similar for the  $3 \times 3 \times 3$ ,  $4 \times 4 \times 4$ , and  $5 \times 5 \times 5$  grids, indicating that all the chosen levels of refinement beyond the coarsest  $2 \times 2 \times 2$  grid are likely to predict sufficiently conservative  $F^{95}$  values and corresponding uncertainty estimates across the test primary space  $\hat{\mathbb{P}}$ .

Estimators developed with the highlighted framework provide powerful tools for establishing uncertainty in coupled FEM-NN predictions. In an application setting, these uncertainty estimators can be used to infer (with some established reliability) the uncertainty scaling factor  $F$  for any load

spanning the primary space. This scaling factor provides an uncertainty range associated with a QoI prediction from an FEM-NN simulation that would likely encompass the true values obtained in FEM simulations.

The cost associated with running FEM-NN simulations includes the time needed to generate training data, the training costs for each BNN, and the time needed to obtain the uncertainty scaling factors and evaluate them for the appropriate load cases. The computational savings gained from using such FEM-NN simulations versus traditional FEM simulations are thus only achieved when the savings in running these simulations overcomes the aforementioned costs. In practice, the process of calibrating an uncertainty estimator and establishing its reliability based on test data points can become prohibitively time-consuming, particularly for FEM simulations involving high-fidelity meshes with millions of elements. In this expensive FEM setting, the selection and discretization of the calibration/test sets in primary space is crucial to obtain reliable uncertainty estimators. This topic will be explored in future work.

## 5. Conclusion

In this article, a method to systematically estimate the uncertainty of NN predictions is introduced within the context of a coupled FEM-NN approach used for predicting the elastic and plastic deformation of solids. Such uncertainty methods are crucial in establishing reliability (or lack thereof) for surrogate modeling approaches whose overall trustworthiness is intimately tied to the bias errors and variance of its predictions.

Ensembles of BNNs are used to enhance diversity in predictions while reducing persistent bias/error. Moreover, sampling of such ensembles is leveraged to quantify the range of possible predictions for a wide variety of test cases that require both interpolation and extrapolation on the space of possible inputs. To establish an uncertainty procedure, a scaling factor is introduced that measures the overall bias-to-variance ratio for possible predictions in the entire test space.

Uncertainty estimators are developed for an example problem where the forces for a subregion of a simple three-element geometry are approximated by ensembles of BNNs that are fed the displacements as inputs. Calibration of this estimator was achieved by illustrating its ability to reliably estimate uncertainty scaling factors for a wide variety of interpolative and extrapolative test cases.

### List of Symbols

$\Omega_{\text{FEM}}$	discretized domain containing all finite elements
$\Omega_{\text{NN}}$	discretized domain containing all NN elements
$[\mathbf{F}_{\text{NN}}^{e.int}]$	element-level internal force vector for NN elements
$[\mathbf{F}_{\text{FEM}}^{e.int}]$	element-level internal force vector for finite elements
$[\mathbf{F}^{int}]$	global internal force vector containing contributions from FEM and NN domains
$\mathbb{P}$	primary space where the set of boundary conditions for a given boundary-value problem are defined
$\mathbb{S}_u$	secondary input space where the NN element displacements are defined
$\mathbb{S}_{ur}$	reduced secondary input space where the NN element displacements are defined in a lower dimensional space
$\mathbb{S}_f$	secondary output space where the NN element forces are defined
$\mathcal{D}_u$	the full set of displacement input data used for training and validation
$\mathcal{D}_f$	the full set of force output data used for training and validation
$\mathcal{B}$	the full set of boundary conditions defined in primary space that are used for training and validation
$F$	the uncertainty scaling factor
$\mathcal{F}$	the piece-wise linear function that estimates the upper bounds of the uncertainty scaling factor for a given set of boundary conditions defined in primary space

- $\mathcal{T}_{\text{calibration}}$  the set of boundary conditions defined in primary space that are used to calibrate/fit the coefficients of the function  $\mathcal{F}$
- $\hat{\mathcal{T}}$  the set of boundary conditions defined in primary space that are used to test the reliability of the uncertainty estimates predicted by the function  $\mathcal{F}$

**Data availability statement.** The data that support the findings of this study are available upon reasonable request. The neural networks developed in this work were generated using PyTorch, available on the following repository <https://github.com/pytorch/pytorch>.

**Acknowledgments.** The authors would like to acknowledge Timothy Shelton and Payton Lindsay for implementing the majority of the machine learning element capability in SIERRA/SM. Additionally, the authors thank Kyle Karlson, Eric Parish, and John Mersch for their invaluable insight and advice on the work presented in this article.

**Author contribution.** Conceptualization: G.L.B., D.M., V.R.; Data curation: G.L.B., D.M.; Formal analysis: G.L.B., D.M., V.R.; Methodology: G.L.B., D.M., V.R.; Software: G.L.B., D.M.; Supervision: V.R.; Validation: G.L.B., D.M., V.R.; Visualization: G.L.B., D.M.; Writing—original draft: G.L.B., D.M.; Writing—review and editing: G.L.B., D.M.

**Funding statement.** This work was supported by the Advanced Simulation and Computing (ASC) program at Sandia National Laboratories. Sandia National Laboratories is a multi-mission 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-NA0003525.

**Competing interest.** The authors declare no competing interests.

## References

- Aldakheel F, Satari R and Wriggers P** (2021) Feed-forward neural networks for failure mechanics problems. *Applied Sciences* 11, 6483.
- Baiges J, Codina R, Castañar I and Castillo E** (2020) A finite element reduced-order model based on adaptive mesh refinement and artificial neural networks. *International Journal for Numerical Methods in Engineering* 121(4), 588–601.
- Balzani D, Gandhi A, Klawonn A, Lanser M, Rheinbach O and Schröder J** (2016) One-way and fully-coupled FE2 methods for heterogeneous elasticity and plasticity problems: Parallel scalability and an application to thermo-elastoplasticity of dual-phase steels. In *Software for Exascale Computing-SPPEXA 2013–2015*. Cham: Springer, pp. 91–112.
- Benabou L** (2021) Development of LSTM networks for predicting viscoplasticity with effects of deformation, strain rate, and temperature history. *ASME Journal of Applied Mechanics* 88, 071008.
- Blundell C, Cornebise J, Kavukcuoglu K and Wierstra D** (2015) Weight uncertainty in neural network. In *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37. Lille: Proceedings of Machine Learning Research, pp. 1613–1622.
- Budarapu PR, Zhuang X, Rabczuk T and Bordas SP** (2019) Multiscale modeling of material failure: Theory and computational methods. *Advances in Applied Mechanics* 52, 1–103.
- Capuano G and Rimoli JJ** (2019) Smart finite elements: A novel machine learning application. *Computer Methods in Applied Mechanics and Engineering* 345, 363–381.
- Duarte CA and Kim DJ** (2008) Analysis and applications of a generalized finite element method with global–local enrichment functions. *Computer Methods in Applied Mechanics and Engineering* 197(6), 487–504.
- Fort S, Hu H and Lakshminarayanan B** (2019) Deep ensembles: A loss landscape perspective. [arXiv:1912.02757](https://arxiv.org/abs/1912.02757).
- Gal Y and Ghahramani Z** (2015) Bayesian convolutional neural networks with Bernoulli approximate variational inference. [arXiv:1506.02158](https://arxiv.org/abs/1506.02158).
- Ganapathysubramanian S and Zabarar N** (2004) Design across length scales: A reduced-order model of polycrystal plasticity for the control of microstructure-sensitive material properties. *Computer Methods in Applied Mechanics and Engineering* 193(45), 5017–5034.
- Geers MGD, Kouznetsova VG and Brekelmans WAM** (2010) Multi-scale computational homogenization: Trends and challenges. *Journal of Computational and Applied Mathematics* 234(7), 2175–2182.
- Hahn GJ and Meeker WQ** (1991) *Statistical Intervals: A Guide for Practitioners*. New York: Wiley & Sons.
- Huang D, Fuhg JN, Weißenfels C and Wriggers P** (2020) A machine learning based plasticity model using proper orthogonal decomposition. *Computer Methods in Applied Mechanics and Engineering* 365, 113008.
- Hughes TJR** (2012) *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. New York: Dover Civil and Mechanical Engineering, Dover Publications.
- Hüllermeier E and Waegeman W** (2021) Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning* 110(3), 457–506.



- Im S, Lee J and Cho M** (2021) Surrogate modeling of elasto-plastic problems via long short-term memory neural networks and proper orthogonal decomposition. *Computer Methods in Applied Mechanics and Engineering* 385, 114030.
- Jolliffe IT and Morgan BJT** (1992) Principal component analysis and exploratory factor analysis. *Statistical Methods in Medical Research* 1(1), 69–95.
- Kalidindi SR** (2015) *Hierarchical Materials Informatics: Novel Analytics for Materials Data*. Amsterdam: Elsevier.
- Karlson KN, Skulborstad AJ, Madison J, Polonsky A, Jin H, Jones A, Sanborn B, Kramer S, Antoun B, and Lu W.** (2023) Toward accurate prediction of partial-penetration laser weld performance informed by three-dimensional characterization – part II:  $\mu$ CT based finite element simulations. *Tomography of Materials and Structures* 2, 100007.
- Kendall A and Gal Y** (2017) What uncertainties do we need in Bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*, vol. 30. Long Beach, CA: Curran Associates, Inc.
- Kingma DP and Ba J** (2014) Adam: A method for stochastic optimization. [arXiv:1412.6980](https://arxiv.org/abs/1412.6980).
- Klawonn A, Lanser M, Rheinbach O and Uran M** (2021) Fully-coupled micro–macro finite element simulations of the Nakajima test using parallel computational homogenization. *Computational Mechanics* 68(5), 1153–1178.
- Knoblauch J, Jewson J and Damoulas T** (2019) Generalized variational inference: Three arguments for deriving new posteriors. [arXiv:1904.02063](https://arxiv.org/abs/1904.02063).
- Knott JF** (1973) *Fundamentals of Fracture Mechanics*. London: Gruppo Italiano Frattura.
- Koeppel A, Bamer F and Markert B** (2018) An intelligent meta-element for linear elastic continua. *PAMM* 18(1), e201800283.
- Koeppel A, Bamer F and Markert B** (2019) An efficient Monte Carlo strategy for elasto-plastic structures based on recurrent neural networks. *Acta Mechanica* 230(9), 3279–3293.
- Koeppel A, Bamer F and Markert B** (2020) An intelligent nonlinear meta element for elastoplastic continua: Deep learning using a new time-distributed residual U-net architecture. *Computer Methods in Applied Mechanics and Engineering* 366, 113088.
- Lakshminarayanan B, Pritzel A and Blundell C** (2017) Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, vol. 30. Long Beach, CA: Curran Associates, Inc.
- LeCun Y, Bengio Y and Hinton G** (2015) Deep learning. *Nature* 521(7553), 436–444.
- Liu J and Marsden AL** (2018) A unified continuum and variational multiscale formulation for fluids, solids, and fluid–structure interaction. *Computer Methods in Applied Mechanics and Engineering* 337, 549–597.
- Miehe C and Bayreuther CG** (2007) On multiscale FE analyses of heterogeneous structures: From homogenization to multigrid solvers. *International Journal for Numerical Methods in Engineering* 71(10), 1135–1180.
- Montes de Oca Zapiain D, Lim H, Park T and Pourboghrat F** (2022) Predicting plastic anisotropy using crystal plasticity and Bayesian neural network surrogate models. *Materials Science & Engineering A* 833, 142472.
- Mota A, Tezaur I and Philpot G** (2022) The Schwarz alternating method for transient solid dynamics. *International Journal for Numerical Methods in Engineering* 123(21), 5036–5071.
- Muhammad W, Brahme AP, Ibragimova O, Kang J and Inal K** (2021) A machine learning framework to predict local strain distribution and the evolution of plastic anisotropy & fracture in additively manufactured alloys. *International Journal of Plasticity* 136, 102867.
- Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, Desmaison A, Kopf A, Yang E, DeVito Z, Raison M, Tejani A, Chilamkurthy S, Steiner B, Fang L, Bai J and Chintala S** (2019) PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, vol. 32. Vancouver, BC: Curran Associates, Inc, pp. 8024–8035.
- Roters F, Eisenlohr P, Hantcherli L, Tjahjanto DD, Bieler TR and Raabe D** (2010) Overview of constitutive laws, kinematics, homogenization and multiscale methods in crystal plasticity finite-element modeling: Theory, experiments, applications. *Acta Materialia* 58(4), 1152–1211.
- Sargsyan K, Najm HN and Ghanem R** (2015) On the statistical calibration of physical models. *International Journal of Chemical Kinetics* 47(4), 246–276.
- SIERRA Solid Mechanics Team** (2021) *Sierra/SolidMechanics 5.2 User's Guide* (SAND Report SAND2021-11231). Albuquerque, NM: Sandia National Laboratories.
- Simo JC and Hughes TJR** (1998) *Classical Rate-Independent Plasticity and Viscoplasticity*. New York, NY: Springer, pp. 71–112.
- Wang D, He H and Liu D** (2018) Intelligent optimal control with critic learning for a nonlinear overhead crane system. *IEEE Transactions on Industrial Informatics* 14(7), 2932–2940.
- Wang K and Sun W** (2018) A multiscale multi-permeability poroplasticity model linked by recursive homogenizations and deep learning. *Computer Methods in Applied Mechanics and Engineering* 334, 337–380.
- Wenyou G and Chang S** (2020) Analysis on block chain financial transaction under artificial neural network of deep learning. *Journal of Computational and Applied Mathematics* 380, 112991.
- Wu PD, MacEwen SR, Lloyd DJ and Neale KW** (2004) Effect of cube texture on sheet metal formability. *Materials Science and Engineering: A* 364(1–2), 182–187.
- Wu L, Nguyen VD, Kilinger NG and Noels L** (2020) A recurrent neural network-accelerated multi-scale model for elasto-plastic heterogeneous materials subjected to random cyclic and non-proportional loading paths. *Computer Methods in Applied Mechanics and Engineering* 369, 113234.
- Wu L and Noels L** (2022) Recurrent neural networks (RNNs) with dimensionality reduction and break down in computational mechanics; application to multi-scale localization step. *Computer Methods in Applied Mechanics and Engineering* 390, 114476.



- Yang L, Meng X and Karniadakis GE** (2021) B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data. *Journal of Computational Physics* 425, 109913.
- Yao Y, Vehtari A, Simpson D and Gelman A** (2018) Using stacking to average Bayesian predictive distributions (with discussion). *Bayesian Analysis* 13(3), 917–1007.
- Zhang A and Mohr D** (2020) Using neural networks to represent von Mises plasticity with isotropic hardening. *International Journal of Plasticity* 132, 102732.
- Zhu Y, Zabaras N, Koutsourelakis PS and Perdikaris P** (2019) Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. *Journal of Computational Physics* 394, 56–81.
- Zou Z, Meng X, Psaros AF and Karniadakis GE** (2022) NeuralUQ: A comprehensive library for uncertainty quantification in neural differential equations and operators. [arXiv:2208.11866](https://arxiv.org/abs/2208.11866).

---

**Cite this article: Bergel GL, Montes de Oca Zapiain D and Romero V** (2023). Neural network ensembles and uncertainty estimation for predictions of inelastic mechanical deformation using a finite element method-neural network approach. *Data-Centric Engineering*, 4, e23. doi:10.1017/dce.2023.17