Mechanics of Materials 184 (2023) 104749

Contents lists available at ScienceDirect

Mechanics of Materials

journal homepage: www.elsevier.com/locate/mecmat

Investigating deep learning model calibration for classification problems in mechanics

Saeed Mohammadzadeh [a], Peerasait Prachaseree [b], Emma Lejeune [b,*]

[a] Division of Systems Engineering, Boston University, Boston, MA 02215, United States of America
[b] Department of Mechanical Engineering, Boston University, Boston, MA 02215, United States of America

ARTICLE INFO

Keywords:
Machine learning
Mechanics
Deep learning
Open science
Benchmark data
Model calibration

ABSTRACT

Recently, there has been a growing interest in applying machine learning methods to problems in engineering mechanics. In particular, there has been significant interest in applying deep learning techniques to predicting the mechanical behavior of heterogeneous materials and structures. Researchers have shown that deep learning methods are able to effectively predict mechanical behavior with low error for systems ranging from engineered composites, to geometrically complex metamaterials, to heterogeneous biological tissue. However, there has been comparatively little attention paid to deep learning model calibration, i.e., the match between predicted probabilities of outcomes and the true probabilities of outcomes. In this work, we perform a comprehensive investigation into machine learning model calibration across 7 open access engineering mechanics datasets that cover three distinct types of mechanical problems. Specifically, we evaluate both model and model calibration error for multiple machine learning methods, and investigate the influence of ensemble averaging and post hoc model calibration via temperature scaling. Overall, we find that ensemble averaging of deep neural networks is both an effective and consistent tool for improving model calibration, while temperature scaling has comparatively limited benefits. Looking forward, we anticipate that this investigation will lay the foundation for future work in developing mechanics specific approaches to deep learning model calibration.

1. Introduction

Over the past decade, there have been unprecedented advances in applying machine learning techniques to problems in mechanics. Researchers have used machine learning approaches to enable design optimization (Gongora et al., 2022; Guo et al., 2021; Hanakata et al., 2020; Shin et al., 2022; Wang et al., 2020a), inverse analysis (Ardizzone et al., 2018; Wang et al., 2019), real-time predictions (Jin et al., 2020; Kapteyn et al., 2021; Zandigohar et al., 2021), and multi-scale modeling (Alber et al., 2019; Karapiperis et al., 2021; Mann and Kalidindi, 2022; Vlassis et al., 2020; Yin et al., 2022) among many other applications. There has also been a growing interest in using machine learning approaches for uncertainty quantification for constitutive modeling (Joshi et al., 2022; Sun et al., 2022) and multi-fidelity surrogate modeling (Han et al., 2022; Perdikaris et al., 2015; Gander et al., 2022). Overall, machine learning models have been repeatedly shown to make predictions about mechanical behavior with low error. However, to date, there has been significantly less investigation into machine learning model *calibration*, i.e., the match between predicted probabilities of outcomes and the true probabilities of outcomes (Gneiting and Raftery, 2007; Guo et al., 2017; Minderer et al., 2021; Naeini et al., 2015; Niculescu-Mizil and Caruana, 2005; Zadrozny and Elkan, 2002).

For applications in engineering design and real world decision making, understanding model calibration alongside model error is essential. In the computational mechanics community, there is a rich history of rigorously studying uncertainty quantification and model calibration (Arendt et al., 2012; Psaros et al., 2022; Wang et al., 2020b). However, for deep learning models applied to problems in mechanics in particular, which tend to have low model error with no associated promise of being well calibrated (Guo et al., 2017), this is a *current knowledge gap*. In Fig. 1, we illustrate the concept of model calibration for binary classification problems. And, in Fig. 1b, we specifically highlight that *low model error* and *low model calibration error* are not necessarily synonymous. Critically, machine learning models can exhibit high accuracy yet suffer from poor calibration. Thus, in this work, our goal is to work towards addressing this knowledge gap by adding additional context specific to deep learning based classification problems in mechanics.

In the machine learning community broadly defined, there is growing interest in improving model calibration without compromising

* Corresponding author.
E-mail addresses: saeedmhz@bu.edu (S. Mohammadzadeh), pprachas@bu.edu (P. Prachaseree), elejeune@bu.edu (E. Lejeune).

https://doi.org/10.1016/j.mechmat.2023.104749
Received 30 November 2022; Received in revised form 6 June 2023; Accepted 13 July 2023
Available online 20 July 2023
0167-6636/© 2023 Elsevier Ltd. All rights reserved.

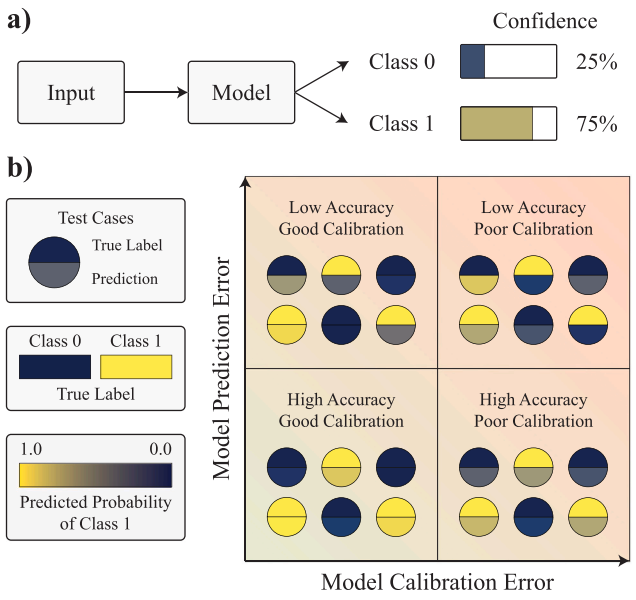

Fig. 1. Conceptual illustration of model calibration for binary classification problems. Panel (a) depicts a standard binary classification task where the model outputs probabilities for both "Class 0" and "Class 1". Panel (b) illustrates the combined implications of both "model error" and "model calibration error" via a toy example. In this graph, the bottom left quadrant represents high accuracy and good calibration and the top right quadrant represents low accuracy and poor calibration. In the circular test cases, the top half of each circle represents the true label, and the bottom half shows the predicted probability for class 1. Overall, low model calibration error is achieved when the network has high confidence for correct predictions and low confidence for incorrect predictions.

predictive accuracy (Guo et al., 2017; Minderer et al., 2021). For deep learning in particular, where model calibration remains poorly understood, there is a concurrent focus on *evaluating* model calibration and on *improving* model calibration. For example, there is growing attention to empirical evaluation of established model calibration metrics (e.g., expected calibration error defined in Section 2.5.3) across different deep learning model architectures (Guo et al., 2017; Minderer et al., 2021). In the computer vision community, researchers have empirically investigated the relationship between model error and emergent model calibration error across multiple architectures on the ImageNet dataset (Minderer et al., 2021; Deng et al., 2009). There has also been significant work towards developing methods specifically for *improving* model calibration (Guo et al., 2017; Lakshminarayanan et al., 2017; Platt et al., 1999; Rahaman and Thiery, 2020; Zadrozny and Elkan, 2002; Zhang et al., 2020), including work that predates the prevalent current interest in deep learning (Gneiting and Raftery, 2007; Niculescu-Mizil and Caruana, 2005). For example, Platt Scaling, a post hoc calibration method where scores of the trained model are additionally trained through logistic regression, was initially developed for support vector machines (Platt et al., 1999). In addition to Platt Scaling, there are multiple post hoc calibration methods that rely on an additional held out training dataset (Zadrozny and Elkan, 2002). For deep learning, deep ensembles (Lakshminarayanan et al., 2017), temperature scaling (Guo et al., 2017), and combinations of deep ensembles and temperature scaling (Rahaman and Thiery, 2020; Zhang et al., 2020) are straightforward and commonly implemented strategies that we will investigate here.

In this work, our goal is to perform a comprehensive investigation into deep learning model calibration for classification problems in mechanics. As our ability to train large deep learning based models with low error becomes more commonplace (Elhassouny and Smarandache, 2019; Guo et al., 2017), working towards better model calibration is a natural next step. The structure of our investigation is informed by two high level objectives. First, because deep learning model performance is *dataset dependant*, it is our goal to design and implement a mechanics-specific challenge for assessing different approaches to model calibration. Namely, we want to create a multi-faceted framework to apply broad advances in machine learning to the mechanics domain. Second, we want to conduct a study that can be directly leveraged by others. This means that we not only want our findings to be of clear utility to others, but also that we want our framework to be directly accessible for others to build on it to assess alternative methods. This structure is directly informed by similar investigations into deep learning methods conducted by others outside the field of mechanics (Kissas et al., 2022; Minderer et al., 2021; Do et al., 2020; Mehrtash et al., 2020).

Following these high level goals, the foundation for our investigation is 7 previously published datasets that span three distinct mechanical problems, detailed in Section 2.1. Necessitated by the diversity in these three mechanical problems, we train distinct problem-specific deep learning models on these datasets, detailed in Section 2.2. And, across all datasets, we explore the influence of ensemble averaging, detailed in Section 2.3, and temperature scaling, detailed in Section 2.4, on model calibration. In Section 3, we present the main findings from our investigation as plots of machine learning model error with respect to machine learning model calibration error. To our knowledge, this is the largest investigation of deep learning model calibration on open access mechanics datasets to date. It is our hope that this investigation is both informative to others, and will lay the foundation for further exploration of this important topic.

2. Methods

In this Section, we will begin by introducing the datasets used in this investigation. Please note that all datasets used in this study have been previously published by our group under Creative Commons Attribution-ShareAlike 4.0 International licenses, and are thus freely available for others to use in follow-up studies to this work. Then, in Section 2.2, we will describe the machine learning models investigated in this work. In Section 2.3 we will describe our implementation of ensemble averaging, and in Section 2.4 we will describe our implementation of temperature scaling. Finally, in Section 2.5, we will specify the error and calibration metrics used to report results in Section 3.

2.1. Benchmark datasets used in this study

In this investigation, our goal is to comprehensively evaluate model calibration on a diverse set of mechanics-based classification datasets. To this end, we will conduct our analysis on 7 open access datasets across three types of mechanical problems. In Section 2.1.1, we provide background details on the "Buckling Instability Classification" (BIC) dataset and sub-datasets (Lejeune, 2020a), in Section 2.1.2, we provide background details on the "Asymmetric Buckling Columns" (ABC) dataset and sub-datasets (Prachaseree and Lejeune, 2022a), and in Section 2.1.3 we provide details on the "Mechanical MNIST – Crack Path" dataset (Mohammadzadeh and Lejeune, 2021). We note briefly that all datasets are derived from simulations conducted via the open source finite element analysis software FEniCS (Alnæs et al., 2015; Logg et al., 2012), and the structures in the ABC dataset are generated through Gmsh (Geuzaine and Remacle, 2009). Overall, these datasets cover both a range of mechanical mechanisms (i.e., both geometric and material nonlinearity), and rely on a range of deep learning techniques (i.e., standard neural networks Lejeune, 2021, graph neural networks Prachaseree and Lejeune, 2022b, and convolutional networks Mohammadzadeh and Lejeune, 2022). Specific details for accessing each dataset and the additional background information required to recreate each dataset are provided in Section 5.

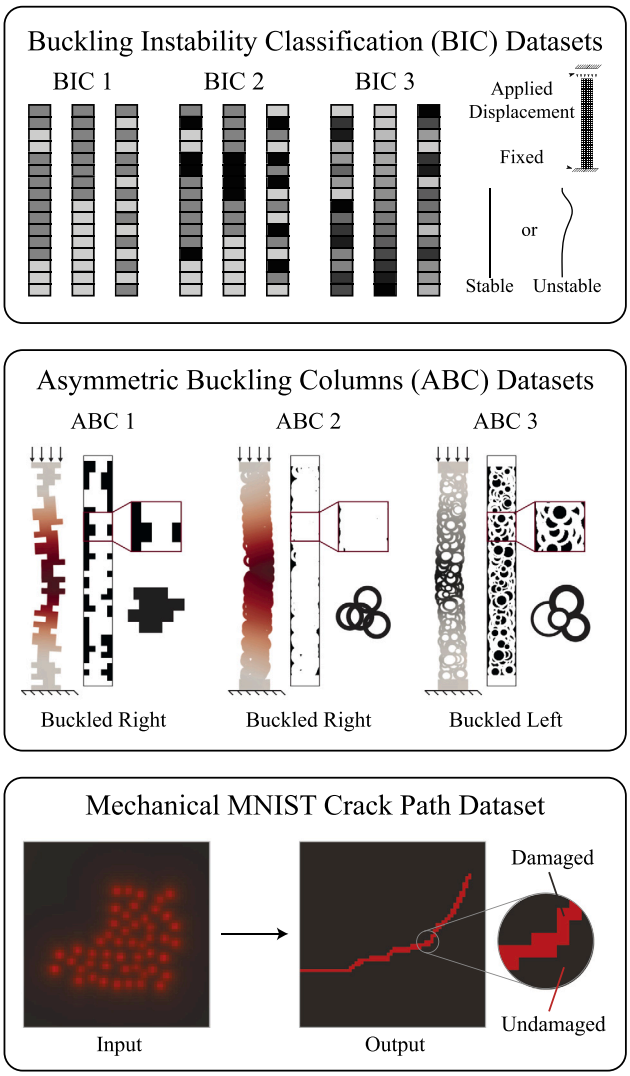

Fig. 2. Schematic illustration of the 7 datasets used in this study: the BIC dataset (which contains BIC 1, BIC 2, and BIC 3) (Lejeune, 2020a), the ABC dataset (which contains ABC 1, ABC 2, and ABC 3) (Prachaseree and Lejeune, 2022a), and the Mechanical MNIST – Crack Path dataset (Mohammadzadeh and Lejeune, 2021).

2.1.1. Buckling Instability Classification (BIC)

The BIC dataset was disseminated in conjunction with our previous publication exploring multiple straightforward approaches to classification problems in mechanics (Lejeune, 2021). The BIC dataset contains three sub-datasets: "BIC 1", "BIC 2", and "BIC 3" that differ only in their input parameter distribution. In all cases, the input is the material property distribution of a heterogeneous column that is then subject to a fixed level of applied compressive displacement. The output then corresponds to class "Stable" or "Unstable" based on the results of a Finite Element Analysis (FEA) simulation. For all sub-datasets, the input property distribution of each sample is represented by a 16×1 vector. For BIC 1, there are two possible discrete modulus E values: $E = 1$, and $E = 4$. For BIC 2, there are three possible discrete values: $E = 1$, $E = 4$, and $E = 7$. For BIC 3, the modulus varies continuously to three degrees of precision in the range $E = [1, 8]$. Further details regarding data curation and our FEA implementation are available in our previous publication (Lejeune, 2021). For the work presented in this manuscript, we used 10,000 samples for machine learning model training and 1,000 samples for post hoc model calibration across all three sub-datasets, and 6,553 samples for BIC 1 and 10,000 samples for BIC 2 and BIC 3 for model testing. Briefly, we note that in Section 3.1, we also specifically investigate training set sizes of 200, 500, 1,000, 2,000, 5,000, 10,000 samples. For the BIC 1 dataset, the ratio between "Stable" and "Unstable" samples is 0.28, for BIC 2 it is 0.40, and for BIC 3 it is 0.26.

2.1.2. Asymmetric Buckling Columns (ABC)

As a follow up to the BIC dataset, we introduced the ABC dataset (Prachaseree and Lejeune, 2022a) in our previous work in conjunction with an exploration of geometric deep learning for mechanics-specific classification problems (Prachaseree and Lejeune, 2022b). Similar to BIC, the ABC dataset contains three subdatasets, where each subdataset corresponds to a different algorithm for generating the geometry of the input domain. For ABC 1, columns are generated by vertically stacking rectangular blocks of randomly varying widths. For ABC 2, columns are generated by randomly overlaying rings of identical inner and outer radii. For ABC 3, columns are generated by overlaying and *trimming* rings of varying inner and outer radii (i.e., varying size and thickness). For all sub-datasets, the columns are subjected to fixed-fixed boundary conditions and are compressed until the onset of buckling. Each input geometry in the ABC dataset is then classified as buckling "left" or "right". Further details of data curation, and FEA implementation are available in our previous publication (Prachaseree and Lejeune, 2022b). For the work presented in this manuscript, we used 20,000 samples for machine learning model training, 1,000 samples for post hoc model calibration, and 2,500 samples for model testing for each of the three sub-datasets. For all three ABC sub-datasets, the classes are balanced.

2.1.3. Mechanical MNIST – Crack path

The Mechanical MNIST dataset collection (Lejeune, 2020b) is a collection of benchmark datasets initially conceptualized as mechanics-relevant drop-in replacements for the popular MNIST dataset (LeCun and Cortes, 2010). For the datasets in the Mechanical MNIST collection, input bitmaps dictate heterogeneous material properties, and outputs are defined as curated results from FEA simulations. The "Mechanical MNIST – Crack Path" dataset is an example from the collection where the input bitmap distribution is a heterogeneous pattern of inclusions derived from the Fashion MNIST dataset (Xiao et al., 2017), and the main output is a damage field predicted by a linear elastic phase-field fracture simulation (Wu et al., 2020; Wu, 2017). For this manuscript, we will focus exclusively on the 64×64 input bitmap and a downsampled 64×64 output crack path. Notably, each pixel in this downsampled crack path is in either the "damaged" (true) class or the "undamaged" (false) class, thus conceptualizing the Mechanical MNIST – Crack Path dataset as a binary classification problem similar to the BIC and ABC datasets described previously. Further details of data curation and our FEA implementation are available in our previous publication (Mohammadzadeh and Lejeune, 2022). In the original dataset there are 60,000 samples in the training set and 10,000 samples in the test set. For the work in this manuscript, we used the first 10,000 samples from the training set for machine learning model training, the next 1,000 samples from the training set for post hoc model calibration, and 10,000 samples from the test set for model testing. For this dataset, classes are heavily imbalanced, where the "damaged" class corresponds to 2.88% of pixels.

2.2. Machine learning models investigated

The main focus of this work is on *deep learning* model calibration. However, in Section 3.1, we provide baseline comparisons to Gaussian Process Classification and Support Vector Classification to add additional context to our results. Here we briefly summarize these methods along with the Neural Network based approaches used for prediction. As a brief note, details for accessing the code to reproduce these models are given in Section 5.

2.2.1. Gaussian process classification

Gaussian Processes are commonly used in machine learning literature for both classification and regression tasks when uncertainty quantification is critical (Bartók et al., 2022). The Gaussian Process Classification (Williams and Rasmussen, 2006), which we use in this work, is a generalization of the linear logistic regression model where the linear latent function is replaced by a Gaussian Process. To train a Gaussian Process Classification in the context of a machine learning problem, the user must define a kernel function that will determine the form of the covariance matrix. For further details on Gaussian Process methods in machine learning, we refer the reader to the literature (Williams and Rasmussen, 2006). In this work, we used scikit-learn (Pedregosa et al., 2011) to train Gaussian Process Classification with a Radial Basis Function kernel (Duvenaud, 2014) on the BIC 1, BIC 2, and BIC 3 datasets. The performance of Gaussian Process Classification on these data is shown in Section 3.1.

2.2.2. Support vector classification

Support Vector Machines are a commonly used machine learning algorithm for classification problems (Hearst et al., 1998). The Support Vector Machine was initially developed for binary classification problems and later on extended to deal with multi-class classification (Weston and Watkins, 1998) and regression tasks (Drucker et al., 1996). Here, we will focus on "Support Vector Classification" for binary classification. In brief, the general idea behind Support Vector Classification is to transform the data in a high dimensional space and identify a hyperplane that most accurately separates the classes. Similar to Gaussian Process Classification, choice of kernel function impacts Support Vector Classification performance in the context of machine learning. In this work, we use scikit-learn (Pedregosa et al., 2011) to train Support Vector Classifications with a Radial Basis Function kernel and no additional regularization on the BIC 1, BIC 2, and BIC 3 datasets. The performance of Support Vector Classifications on these data is shown in Section 3.1. Critically, we note that scikit-learn uses Platt scaling, described in Platt et al. (1999), coupled with five-fold cross-validation to obtain probabilistic outputs from otherwise non-probabilistic Support Vector Classification scores.

2.2.3. Fully Connected Neural Network

Fully Connected Neural Networks are a well-established method for both regression and classification tasks. These networks commonly consist of an input layer, an output layer, and a series of fully connected hidden layers. Each layer applies a linear transformation followed by a non-linear activation function such as Rectified Linear Units (ReLU) (Agarap, 2018) on its input vector. These layers are designed to eventually transform a given input vector into an output vector of the desired size. For the BIC datasets, we use Fully Connected Neural Networks (simply referred to as neural networks) with a 16 node input layer, three 200 node hidden layers, and a 2 node output layer (see Appendix C). The 2 output nodes are the logits indicating to which class "stable" or "unstable" a sample belongs. To reduce overfitting during training, we add batch normalization (Ioffe and Szegedy, 2015) before applying each ReLU activation function and use dropout (Srivastava et al., 2014) with the rate of 0.5 before the second and third hidden layers and output layer. We use the PyTorch library (Paszke et al., 2019) for our implementation, and train each network for 50 epochs using the Adam optimizer (Kingma and Ba, 2014) with an initial learning rate of 0.001, which is dropped to 0.0005 after 25 epochs. We train this model on the BIC 1, BIC 2, and BIC 3 datasets, and present the results in both Sections 3.1 and 3.2.

2.2.4. Graph Neural Network

By design, the ABC datasets contain complex geometries that are attractive to represent as spatial graphs rather than as "image-like" arrays. In our previous work, we identified a set of best performing models and data representation approaches for each ABC subdataset that we directly build on for this work (Prachaseree and Lejeune, 2022b). In brief, each ABC input geometry is first represented as a spatial graph. In our previous work, we identified a high performing strategy for spatial graph representation where spatial graphs are constructed via discretizing the structure into nodes and then performing a ball query to form edges. Based on our prior investigation, ABC 1 had a "medium" node density ($\approx$ 306 nodes per structure) with a ball radius of 40% of the column width, and ABC 2 and ABC 3 have a "dense" node density ($\approx$ 566 and $\approx$ 768 nodes per structure respectively) with a ball radius of 30% of the column width (Prachaseree and Lejeune, 2022b). Given this spatial graph representation, we used PointNet++ layers (Qi et al., 2017) as spatial graph convolution layers coupled with batch normalization (Ioffe and Szegedy, 2015), followed by skip connections and a linear classifier to construct our machine learning model (see Appendix C). To improve model performance, we also augment our dataset by flipping the columns along the x axis, y axis, and both axes while changing labels as needed. All models are implemented with the Pytorch Geometric library (Fey and Lenssen, 2019) and trained using the Adam optimizer (Kingma and Ba, 2014) for 50 epochs. We present the results from this model on the ABC datasets in Section 3.2.

2.2.5. UNet neural network

To complement the models described in Sections 2.2.1–2.2.4 which are trained to predict a single quantity of interest, we train a deep neural network on the Mechanical MNIST – Crack Path dataset that is designed to predict full-field quantities of interest, specifically the whole domain damage field. In our previous work (Mohammadzadeh and Lejeune, 2022), we used a modified version of the UNet model (Ronneberger et al., 2015), the MultiRes-WNet, combined with a convolutional autoencoder for an end-to-end prediction of 256 $\times$ 256 images of the damage field from 64 $\times$ 64 material distribution input images. Here, we regenerated lower resolution output damage fields directly from our FEA results as 64 $\times$ 64 arrays and used a standard UNet with three downsampling and upsampling steps (Siddique et al., 2021). The outputs of the model are logits in the form of two-channel images that can be transformed into probabilities by applying a softmax function to each pixel (see Appendix C). We briefly note that we trained the network by minimizing the Dice-loss (Jadon, 2020). We use the PyTorch library (Paszke et al., 2019) for the UNet model implementation, and train each network for 50 epochs using the Adam Optimizer (Kingma and Ba, 2014). We present the results from this model on the Mechanical MNIST – Crack Path dataset in Section 3.2.

2.3. Ensemble methods

For the Neural Network approaches introduced in Sections 2.2.3–2.2.5, the behavior of each trained neural network will vary based on the random weight initialization. Thus, it is possible to train multiple neural networks and subsequently combine them into an ensemble (Ciregan et al., 2012; Lakshminarayanan et al., 2017). Here, we take a straightforward approach and individually train 10 models with different initialization seeds before aggregating the predictions using soft voting, also referred to as unweighted model averaging (Lakshminarayanan et al., 2017). In soft voting, the predicted probability for each class is averaged over all models and the label with the highest probability then becomes the final class prediction. The goal of ensemble averaging is to increase the overall prediction accuracy. Additionally, if the neural networks are trained with proper scoring rules like cross entropy, ensemble averaging may also lead to averaged probabilities that are well calibrated (Lakshminarayanan et al., 2017). One major goal of this work is to critically evaluate the efficacy of neural network ensemble averaging for deep learning approaches to classification problems in mechanics.

2.4. Post hoc calibration via temperature scaling

Post hoc calibration of neural networks using a held-out calibration dataset is a popular approach with both parametric (e.g., Platt scaling Platt et al., 1999, temperature scaling, matrix scaling Guo et al., 2017) and non-parametric (e.g., Bayesian Binning Naeini et al., 2015, isotonic regression Zadrozny and Elkan, 2002) implementations. Motivated by its popularity in the literature, we choose temperature scaling as a standard post hoc calibration technique. Specifically, given a trained classifier, we divide the logits vector $\mathbf{z}$ by a single variable T called the temperature. The optimal temperature is obtained by minimizing the Negative Log Likelihood (NLL) on the held out calibration set. The NLL is written as:

$$\min_{T} \quad \sum_{i=1}^{N_c} \mathrm{NLL}\Big(\sigma(\mathbf{z}_i / T),\, y_i\Big) \qquad \text{s.t.} \quad T > 0 \tag{1}$$

where $\sigma(\mathbf{x})$ is the softmax function, y is the true labels, N_c is the number of sample points in the calibration set, and $\mathbf{z}$ is the previously defined logits vector. Notably, temperature scaling can be applied either before or after ensemble averaging (Rahaman and Thiery, 2020).

In Section 3.2, we report the results of applying post hoc calibration methods on our datasets. For clarity, the methods investigated are defined as follows:

- *Method **I**:* Individual neural network *without* post hoc calibration.
- *Method **I-C**:* Individual neural network *with* post hoc calibration via temperature scaling.
- *Method **E-M1**:* Ensemble neural network *without* post hoc calibration.
- *Method **E-M2**:* Ensemble neural network *with* post hoc calibration via temperature scaling applied *before* ensemble averaging.
- *Method **E-M3**:* Ensemble neural network *with* post hoc calibration via temperature scaling applied *after* ensemble averaging.

Fig. 6 in Section 3.2 and Fig. 8 in Appendix B directly reference these definitions.

2.5. Error and calibration metrics reported in this investigation

In this work, all supervised learning tasks are binary classification problems. For the datasets without severely imbalanced class labels (i.e. BIC and ABC) we report classification accuracy as our error metric, while for the severely imbalanced class label (i.e. damaged pixels in the Mechanical MNIST Crack Path dataset) we report the F_1 score as our error metric. To measure calibration, we report the Expected Calibration Error (ECE). In all cases, we report these metrics on our held out test datasets. Details on metrics for model error and model calibration error are as follows.

2.5.1. Classification error definition for BIC and ABC

We evaluate model performance for the BIC and ABC datasets via traditional classification error. Specifically, we define classification error as the fraction of wrong predicted labels with respect to the total number of labels. Mathematically, this is written as:

$$\mathrm{Error}(y, \hat{y}) = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}(\hat{y}_i \neq y_i) \tag{2}$$

where N is the number of labels to evaluate, y and $\hat{y}$ are the true and predicted labels respectively, and $\mathbf{1}(\hat{y} \neq y)$ represents the 0–1 loss function (Shalev-Shwartz and Ben-David, 2014).

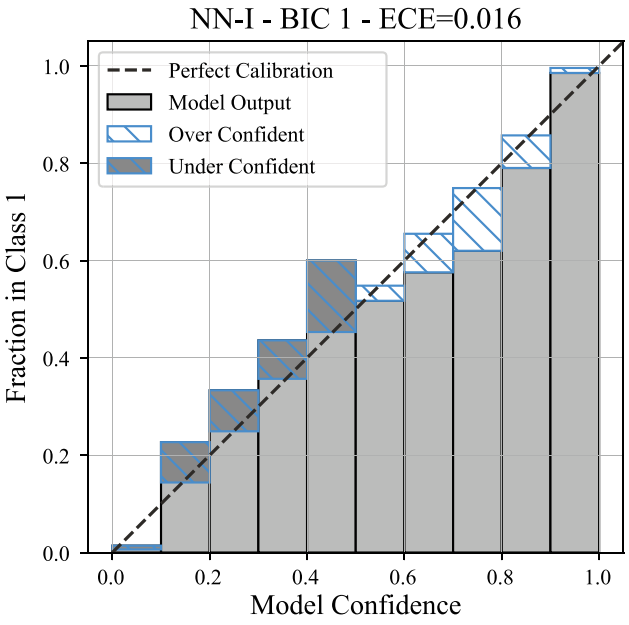

Fig. 3. Representative reliability diagram for an individual neural network trained on the BIC 3 dataset. To promote completeness while preserving data visualization clarity, we report additional reliability diagrams in Appendix B as a supplement to Section 3.

2.5.2. Classification error definition for mechanical MNIST crack path

Following our previous work, the damage field for each sample in the Mechanical MNIST dataset is treated as a binary matrix (image) where 1 represents a damaged sub-region (pixel) and 0 represents an undamaged sub-region (pixels) (Mohammadzadeh and Lejeune, 2022). Because the damaged region represents a crack path, the relative prevalence of damaged pixels is small (2.88%) leading to severe class imbalance. As such, the classification error as defined in Eq. (2) will automatically appear as a small value due to the high percentage of true negative pixels in each prediction. To better evaluate model performance with these imbalanced labels, we report the error as the Sørensen–Dice index, often referred to as the F_1 score, defined as:

$$F_1 = \frac{2\,\text{True Positive}}{2\,\text{True Positive} + \text{False Positive} + \text{False Negative}} \tag{3}$$

where True Positive, False Positive, and False Negative denote the number of correctly predicted damaged pixels, incorrectly predicted undamaged pixels, and incorrectly predicted damaged pixels respectively. We note that F_1 score defined in Eq. (3) is not influenced by the number of true negative pixels (correctly predicted undamaged pixels), making it an easier-to-interpret metric of model predictive performance for the Mechanical MNIST – Crack Path dataset. In Fig. 6, where we report F_1 score, we plot error as $1 - F_1$ on the y axis to maintain visual consistency.

2.5.3. Expected Calibration Error (ECE)

In this work, we use the Expected Calibration Error (ECE) to evaluate model calibration. As stated in Section 1, calibration refers to the match between predicted probabilities of outcomes and the true probabilities of outcomes. For example, when a model is perfectly calibrated, 100 predictions with 80% confidence should be correct 80/100 times. While there are many potential metrics used to evaluate calibration (Guo et al., 2017; Minderer et al., 2021; Naeini et al., 2015; Niculescu-Mizil and Caruana, 2005; Nixon et al., 2019; Ovadia et al., 2019; Zhang et al., 2020), the ECE is one of the most prevalent in the literature and one of the most interpretable. Namely, the ECE is connected to the reliability diagram, a common approach to visualizing model calibration illustrated in Fig. 3 (Guo et al., 2017; Naeini et al., 2015; Niculescu-Mizil and Caruana, 2005). Reliability diagrams are constructed in two steps. First, sample prediction confidences

(i.e., confidence that a given sample is in a chosen class) are binned. In this work, we use 10 equally spaced bins to construct all reliability diagrams. Then, the average bin confidence is compared to the true fraction of samples with the chosen class. The ECE is then computed as the weighted average of the gap between perfect calibration and model confidence within each bin. Mathematically, this is defined as:

$$\mathrm{ECE} = \sum_{i=1}^{B} \frac{n_i}{N} \mid F_i - C_i \mid \tag{4}$$

where B is the number of bins, n_i is the number of samples in each bin, N is the total number of samples, F_i is the frequency of the chosen class in the bin, and C_i is the average confidence that the sample is in the chosen class in the bin. Following this definition, a lower ECE corresponds to a better calibrated model, and a model with ECE= 0 corresponds to a perfectly calibrated model. As indicated in Fig. 3, the reliability curve bins in a perfectly calibrated model will follow the diagonal $y = x$.

We report ECE as our main model calibration metric because it is both prevalent in the literature and relatively interpretable. However, it is not without limitations. For example, ECE is known to be sensitive to the selection of binning scheme (Nixon et al., 2019; Ovadia et al., 2019; Zhang et al., 2020). More specifically, ECE values can depend on the bin size as well as the number of samples in each bin. Multiple modifications to the definition of ECE defined in Eq. (4) such as adaptive binning schemes (Nixon et al., 2019), using the ℓ_2 norm instead of the ℓ_1 norm to compute the ECE (Minderer et al., 2021; Nixon et al., 2019), and kernel-density based methods (Zhang et al., 2020), have been proposed in the literature. Aside from the ECE, other methods that are related to reliability diagrams like the Maximum Calibration Error (Naeini et al., 2015) have been proposed. Alternatively, proper scoring rules (Gneiting and Raftery, 2007) with roots in statistical analysis like the Negative Log-Likelihood (NLL) and Brier Score (Minderer et al., 2021; Ovadia et al., 2019) have been used to evaluate model calibration. However, it is not clear if these proposed methods are significantly better than the standard method for computing ECE, and these new metrics potentially lose some of the clear relationship to the interpretable reliability diagram. Looking forward, we anticipate that the framework we establish in this paper could also be used to investigate the behavior of these alternative metrics. However, this is beyond the scope of our current work.

3. Results and discussion

In Section 2, we introduced 7 datasets (BIC 1, BIC 2, BIC 3, ABC 1, ABC 2, ABC 3, and Mechanical MNIST – Crack Path), described multiple machine learning models for making predictions with these datasets, procedures for ensemble averaging and temperature scaling, and metrics for evaluating model error and calibration. Here, we will begin in Section 3.1 by comparatively evaluating different machine learning methods on the BIC datasets. Then, in Section 3.2, we will investigate multiple strategies for improving model calibration across the BIC, ABC, and Mechanical MNIST – Crack Path datasets. Throughout this Section, we present the results of our study following the format introduced in Fig. 1 where each individual trained machine learning model is represented as a single marker on a prediction error vs. model calibration error axis.

3.1. Evaluating model calibration

Our first major motivation for performing this investigation is that large deep neural networks are prone to being poorly calibrated to an extent that is not well understood (Guo et al., 2017). Thus, by *evaluating* deep neural network model calibration for our mechanics-based datasets, we will make progress towards our general understanding of model calibration as a potentially *emergent* phenomena. In Fig. 4, we plot model error vs. model expected calibration error (ECE) for multiple machine learning models trained on the BIC 1, BIC 2, and BIC 3 datasets. Note that each marker corresponds to the test performance of a trained machine learning model. From Fig. 4, we can directly compare the performance of Gaussian Process Classification and Support Vector Classification with Platt scaling to the performance of neural networks.

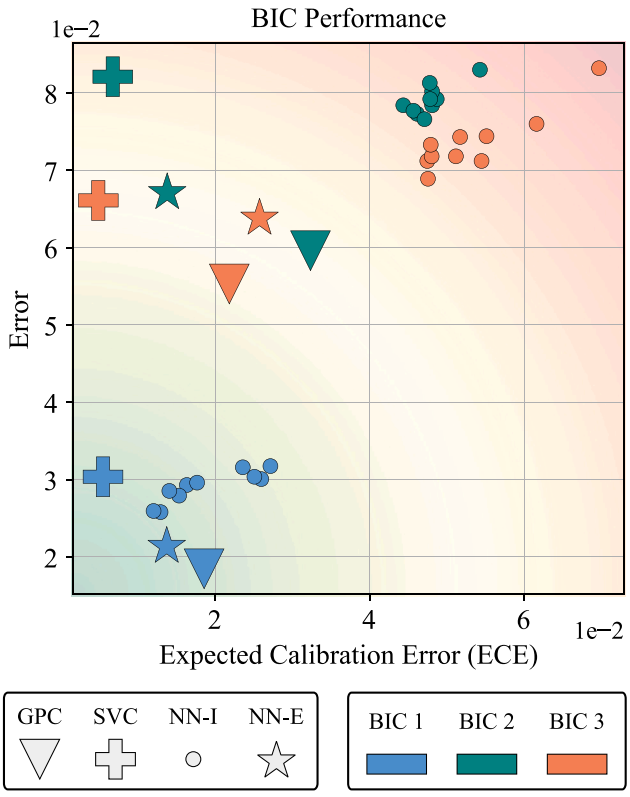

Fig. 4. Error vs. ECE plots for Gaussian Process Classification (GPC), Support Vector Classification (SVC), 10 individual neural networks (NN-I), and an ensemble of 10 neural networks (NN-E) trained on the BIC 1, BIC 2, and BIC 3 datasets. Lower ECE and error indicate better performance (bottom left corner, represented by the green background gradient). Fig. 7 in Appendix B contains reliability diagrams that supplement these results.

Consistent with results from our prior publication (Lejeune, 2021), model performance across different BIC sub-datasets varies due to the different input parameter space for each BIC dataset. A machine learning problem with larger input parameter space (i.e. BIC 3) will typically result in a higher error than a machine learning problem with smaller input parameter space (i.e. BIC 1). Thus, we will make direct comparisons within each dataset, and note overall trends that are consistent across all three datasets. In Fig. 4, we consistently see that individual neural networks have a large ECE range, and tend to be poorly calibrated in comparison to the Gaussian Process Classification and Support Vector Classification models. However, we also consistently see that ensemble neural networks have both lower error and lower ECE, and perform similarly to their Gaussian Process Classification and Support Vector Classification counterparts. This context is important because it justifies the choice of ensemble neural networks as an approach to designing well calibrated deep learning based model frameworks. In Appendix B, Fig. 7, we show supplementary reliability diagrams that further support these results. Overall, this first investigation leads to the expected result that ensemble averaging consistently improves the performance of our deep neural networks. And, it leads to the less expected result that ensemble neural networks have similar calibration performance to the Gaussian Process Classification and Support Vector Classification baselines.

In Fig. 5, we plot model error vs. model expected calibration error (ECE) for multiple model types with different training set sizes trained

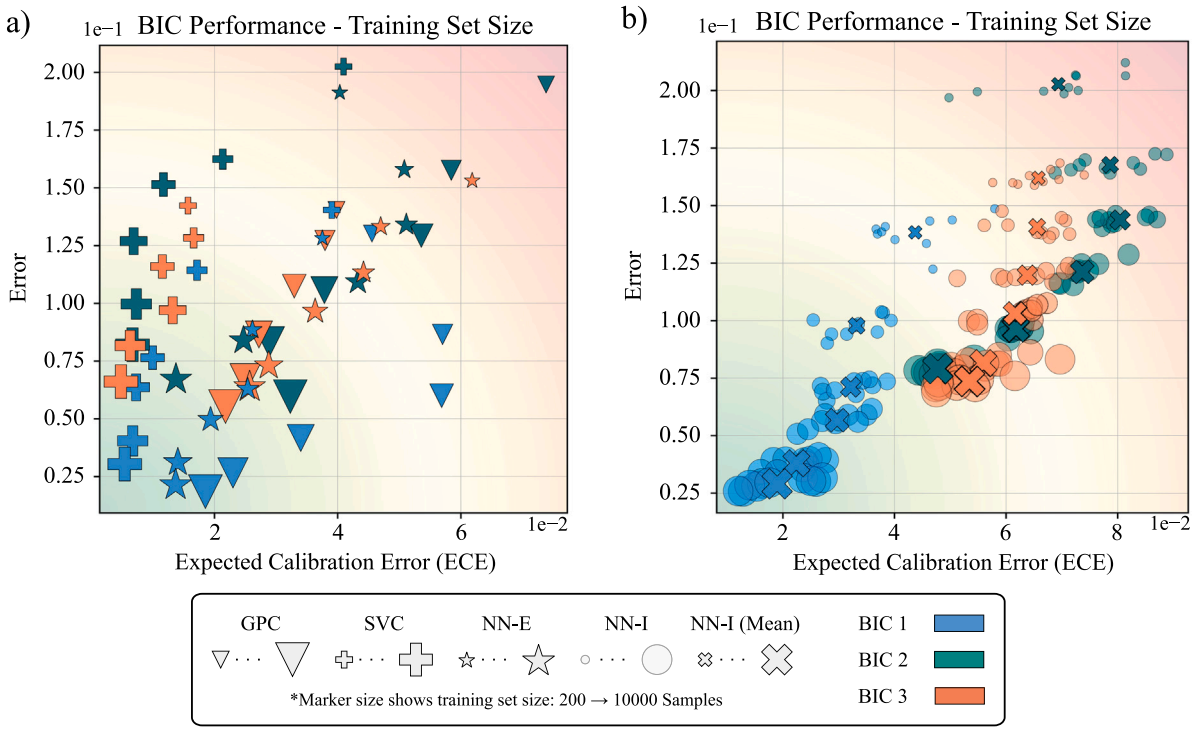

Fig. 5. Error vs. ECE plots with varied training set sizes: (a) Gaussian Process Classification (GPC), Support Vector Classification (SVC), and an ensemble of 10 neural networks (NN-E); (b) 10 individual neural networks (NN-I) and mean values of these 10 networks (NN-I (Mean)). All models are trained on multiple subsets of the BIC 1, BIC 2, and BIC 3 datasets with varying training set sizes (200, 500, 1,000, 2,000, 5,000, and 10,000 samples). Lower ECE and error indicate better performance (bottom left corner, represented by the green background gradient). The Pearson correlation coefficients for error and ECE on BIC 1, BIC 2, and BIC 3, respectively, are as follows. SVC: (0.899, 0.887, 0.917); GPC: (0.691, 0.967, 0.995); NN-E: (0.981, 0.731, 0.983); NN-I: (0.850, 0.612, 0.659); NN-I (Mean): (0.969, 0.656, 0.936).

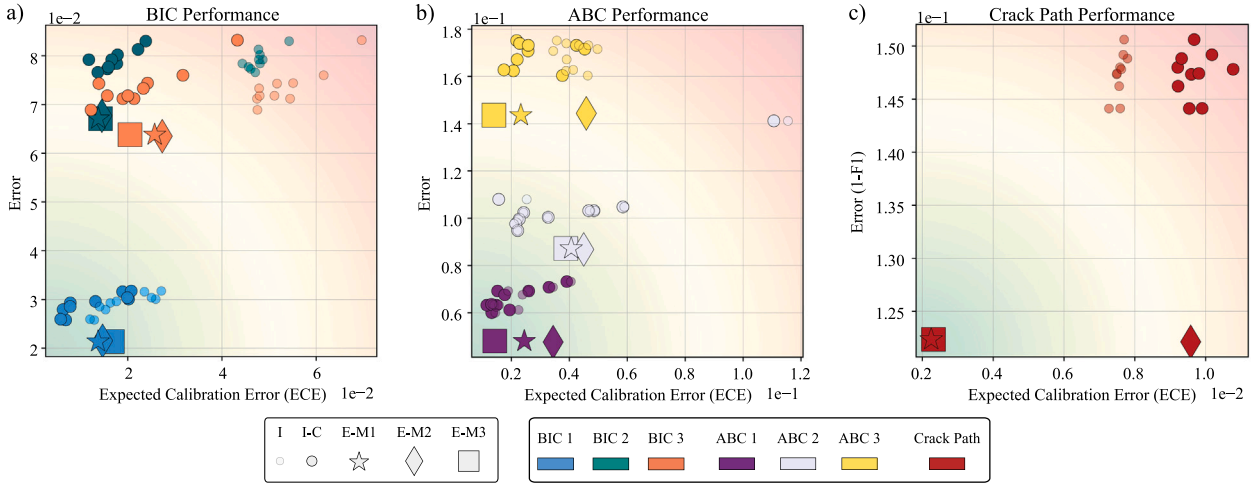

Fig. 6. Visualization of the influence of ensemble averaging and post hoc model calibration on error and ECE for (a) the BIC datasets, (b) the ABC datasets, and (c) the Mechanical MNIST Crack Path dataset. Color indicates the (sub)dataset, and marker style specifies the ensemble averaging and calibration approach, see Section 2.4. Lower ECE and error indicate better performance (bottom left corner, represented by the green background gradient). Fig. 8 in Appendix B contains reliability diagrams that supplement these results.

on the BIC 1, BIC 2, and BIC 3 datasets. In Fig. 5, each marker corresponds to the test performance of a trained machine learning model, and marker size indicates the size of the training dataset. As expected, Fig. 5 shows that error generally decreases as the training set size increases, but eventually reaches a point of "diminishing returns", where additional training data only marginally improves the accuracy. This result is consistent with the standard observation in machine learning and highlights the importance of selecting an appropriate training set size. Beyond this standard result, Fig. 5 illustrates two important results. First, it is clear from the distribution of results across all machine learning models and datasets that model error and ECE are at most weakly correlated (the overall correlation coefficient across all models is 0.516). Namely, low model error does not necessarily indicate low ECE, and high model error does not necessarily indicate high ECE. Second, if we examine individual model types (i.e., Gaussian Process Classification, Support Vector Classification, or ensemble neural networks) and individual datasets (i.e., BIC 1, BIC 2, BIC 3) increasing the training set size consistently lowers both model error and model ECE. For example, for ensemble neural networks the correlation coefficients relating error and ECE are 0.981, 0.731, and 0.983 for BIC 1, BIC 2, and BIC 3 respectively. Critically, for these examples, increasing the training set size improves both model error and model calibration

which is an important observation because it is an actionable strategy for improving both dimensions of performance.

3.2. Improving model calibration

In Sections 2.3 and 2.4, we introduced two strategies for explicitly *improving* model calibration. The first, ensemble averaging, relies on training multiple neural networks on the same data using different random weight initialization. The second, post hoc calibration via temperature scaling, relies on reserving additional data for calibration. In this Section, we will compare the performance of individual neural networks (I), individual neural networks with post hoc calibrated via temperature scaling (I-C), ensemble averaging (E-M1) and two different methods for ensemble averaging combined with post hoc calibrated via temperature scaling (E-M2, and E-M3, see Section 2.4 for definitions). Similar to machine learning literature where it is typical to perform these investigations across multiple datasets, we will investigate all approaches across the BIC, ABC, and Mechanical MNIST – Crack Path datasets in order to identify outcomes that are potentially consistent across diverse types of mechanical data.

Consistent with the results presented in Section 3.1, we find that ensemble averaging without post hoc calibration (E-M1) improves error and ECE across all datasets. We note briefly that in Fig. 6a, which represents the BIC datasets, the I and E-M1 data points are repeated from Fig. 4, and in Fig. 6b, which represents the ABC datasets, the I and E-M1 data points are repeated from our previous publication (Prachaseree and Lejeune, 2022b). Across all datasets shown in Fig. 6a–c, direct comparison of the I and E-M1 points convincingly indicates that ensemble averaging holds up as a strategy for improving model performance and calibration. As outlined in Section 2.3, we would like to emphasize that during the training of each individual network, the weights were randomly initialized. The impact of this random initialization can be observed in Fig. 6 where individual neural networks clearly exhibit variable final performance. One reason for the success of ensemble averaging for improving performance is that it can both mitigate and leverage the downstream effects of random weight initializations. Specifically, random weight initializations mean that the same model inputs may correspond to different model outputs across the individual neural networks. By averaging these predictions, we can not only mitigate poor predictions, but also improve the average efficacy of inconsistent predictions. These positive results, all without post hoc calibration, serve as a baseline for evaluating the efficacy of methods designed specifically to improve calibration such as temperature scaling.

In Fig. 6, we also show the results of three approaches to performing post hoc model calibration via temperature scaling (I-C, E-M2, and E-M3), where the individual models (I) and models with straightforward ensemble averaging (E-M1) serve as the baseline. From Fig. 6, we see that unlike the results of applying ensemble averaging, the results of applying temperature scaling are much less consistent. In contrast with machine learning literature (Guo et al., 2017), when applying temperature scaling to individual neural networks (i.e., comparing the I models to the I-C models), there were limited and inconsistent benefits. For example, in Fig. 6b, temperature scaling leads to limited decreases in ECE for individual networks, while in Fig. 6c temperature scaling leads to a modest increase in ECE. One possible explanation for this difference is that Fig. 6c corresponds to the Mechanical MNIST – Crack Path dataset, which is inherently very unbalanced – there are many fewer "damaged" pixels than "undamaged" pixels. Though we cannot strictly say that this is causal, we can resolutely say that it is important to evaluate methods for improving model calibration on an example of an unbalanced dataset prior to drawing general conclusions. Overall, in comparing individual models with temperature scaling (I-C) to ensemble models without temperature scaling (E-M1), we note that the E-M1 models more consistently lead to both lower ECE and lower error.

When comparing ensemble averaging combined with post hoc calibration via temperature scaling (E-M2 and E-M3) to the baseline of straightforward ensemble averaging (E-M1), we find similar inconsistent results. Specifically, ensemble averaging combined with temperature scaling (E-M2, E-M3) lead to inconsistent performance improvements in comparison to ensemble averaging alone (E-M1). Of note, temperature scaling prior to ensemble averaging (E-M2) led to strikingly worse performance compared to ensemble averaging alone (E-M1) for the Mechanical MNIST – Crack Path dataset, illustrated in Fig. 6c. As stated previously, this dramatic difference may be due to the severe class imbalance present in the Mechanical MNIST – Crack Path dataset. Overall, we found that for these datasets ensemble averaging offers much more consistent performance improvements than post hoc model calibration via temperature scaling. The supplementary reliability diagrams shown in Appendix B, Fig. 8 also support these results. In addition, it is worth mentioning that results shown in Fig. 5, where increasing the training set size led to improvements in both model error and model ECE for ensemble neural networks, also indicate that increasing the initial training set size may be a better use of data resources than post hoc calibration via temperature scaling. However, we acknowledge that this statement may vary based on specific desired outcomes and data resources.

Overall, we note that our findings are based on *empirical* evidence obtained through analyzing the 7 datasets introduced in this manuscript. The complex and presently "black box" nature of deep neural networks means that we should not make either sweeping generalizations or causality claims based on these findings alone. Therefore, for full transparency and completeness, it is important to acknowledge that the trends observed in this study are not necessarily guaranteed to be consistent with either previous or forthcoming literature that is based on different data. For example, others have empirically shown that temperature scaling applied after ensemble averaging (M3) can consistently improve model calibration compared to ensemble averaging alone (M1) (Rahaman and Thiery, 2020). However, because there is no real consensus on the best method for calibrating deep learning models, and calibration strategies to date appear dependent on the type of dataset and deep learning architecture used (Zhang et al., 2020; Guo et al., 2017; Lakshminarayanan et al., 2017; Minderer et al., 2021; Ovadia et al., 2019; Rahaman and Thiery, 2020), we assert that this work is a necessary and important step forward. As such, rather than taking the results of this investigation at face value, we hope that our work (1) highlights the need for more research towards understanding deep neural network calibration, and (2) emphasizes the need for additional domain specific open access datasets for systematically exploring the efficacy of deep learning approaches.

4. Conclusion

To the author's knowledge, this is the largest investigation to date of deep learning model calibration for classification problems in mechanics. From this investigation, we found four key results. First, we found that ensemble neural networks perform comparably to Gaussian Process Classification and Support Vector Classification with Platt scaling in terms of model error and model expected calibration error (ECE) for all three BIC datasets. Second, we found that increasing the training set size decreases both model error and model ECE for all three BIC datasets. Third, we found that ensemble averaging consistently improves both model error and model ECE for all 7 datasets. Fourth, we found that temperature scaling offers limited benefits in comparison to ensemble averaging for all 7 datasets. In summary, the most important result from this study is that ensemble averaging of deep neural networks is both an effective and consistent tool for improving model calibration for problems in mechanics, while temperature scaling has comparatively limited benefits. Overall, we believe that this work demonstrates the utility of large scale studies of machine learning methods applied to problems in mechanics.

Looking forward, we anticipate several major areas of future investigation by both us and others. First, these datasets can be used to investigate alternative approaches to simultaneously improving both model error and ECE. For example, there are multiple approaches to post hoc model calibration beyond temperature scaling that are amenable to similar investigation (Kuleshov et al., 2018; Naeini et al., 2015; Rahimi et al., 2020). Alternatively, Bayesian methods (Kendall and Gal, 2017; Maddox et al., 2019; Zhang and Garikipati, 2021) and evidential deep learning models (Amini et al., 2020; Sensoy et al., 2018) aim to output calibrated predictions without any additional post hoc training. And, building on exciting recent work (Raissi et al., 2019; Yang and Perdikaris, 2019), we anticipate that there are rich possibilities for physics-informed approaches to this problem. Second, these datasets can be used to investigate alternative approaches to evaluating model calibration. As stated in Section 2.5.3, developing more effective metrics remains an open area of research and one that deserves attention in a mechanics-specific context. Third, there is a need to extend this study to additional open-access mechanics-based datasets from diverse sources. Ultimately, we acknowledge that the current study is limited to data generated through FEA, which introduces a potential bias. This highlights the need for further research to explore the problem with data from other sources, such as experimental testing or molecular dynamic simulations, which may contain stochastic behavior. Additionally, our study is limited to simulated data since, to our knowledge, there are currently no open access experimental mechanics datasets that are both amenable to being formulated as a classification problem and sufficiently large to include in this study. We view the lack of experimental data as the biggest limitation of this work. That being said, our hope is that this work both offers a starting point for researchers beginning work with deep learning model calibration, and motivates future mechanics-specific advances in deep learning model calibration. Because all datasets and codes associated with this manuscript are available under open-source licenses, others can readily build on our work and make direct comparisons to alternative methods and datasets.

5. Additional information

All datasets used in this investigation have been previously published in conjunction with prior manuscripts from our group (Lejeune, 2021; Prachaseree and Lejeune, 2022a; Mohammadzadeh and Lejeune, 2022). Each dataset contains both the metadata to interpret files and the code needed to reproduce all results. In all cases, data is shared under a CC BY-SA 4.0 License through the OpenBU Institutional Repository and code is shared under a MIT License through GitHub. The datasets used are as follows:

- **Buckling Instability Classification (BIC)** (Lejeune, 2020a): rectangular columns with heterogeneous material properties are subject to a fixed level of applied displacement and classified as either stable or unstable (i.e., buckled).
  BIC contains three *independent* sub-datasets where all three 16×1 input patterns are sampled from different distributions:

  BIC 1: 16×1 pattern from distribution 1, sampling 2 discrete values (input) $\mapsto$ stable vs. unstable (output)
  BIC 2: 16×1 pattern from distribution 2, sampling 3 discrete values (input) $\mapsto$ stable vs. unstable (output)
  BIC 3: 16×1 pattern from distribution 3, sampling continuous range (input) $\mapsto$ stable vs. unstable (output)

- **Asymmetric Buckling Columns (ABC)** (Prachaseree and Lejeune, 2022a): heterogeneously architected and asymmetric columns with homogeneous material properties are subject to a fixed level of applied displacement and classified as either left buckling or right buckling.
  ABC contains three *independent* sub-datasets where all three input domain architecture types are generated through different procedural approaches:

  ABC 1: spatial graph that represents domains from domain type 1, block stacking (input) $\mapsto$ left vs. right (output)
  ABC 2: spatial graph that represents domains from domain type 2, uniform rings (input) $\mapsto$ left vs. right (output)
  ABC 3: spatial graph that represents domains from domain type 3, clipped non-uniform rings (input) $\mapsto$ left vs. right (output)

- **Mechanical MNIST – Crack Path** (Mohammadzadeh and Lejeune, 2021): two-dimensional square domains with heterogeneous material properties and a defined initial crack are subject to a fixed level of applied displacement. Under these loading conditions, a crack propagates throughout the domain with the crack path dictated by the heterogenous material property distribution. Mechanical MNIST – Crack Path is a single dataset:

  Mechanical MNIST – Crack Path: 64×64 material property array (input) $\mapsto$ 64×64 damage field (output)

All datasets are schematically illustrated in Fig. 2 for a total of 7 independently trained and tested cases (Lejeune, 2020a; Prachaseree and Lejeune, 2022a; Mohammadzadeh and Lejeune, 2021).

The code to reproduce all computational results presented in this paper is available through GitHub (https://github.com/saeedmhz/model-calibration) with the exception of the code to implement the Graph Neural Network described in Section 2.2.4 which is published in the GitHub repository accompanying our previous publication (Prachaseree and Lejeune, 2022b).

CRediT authorship contribution statement

Saeed Mohammadzadeh: Conceptualization, Methodology, Formal analysis, Data curation, Writing – original draft, Writing – review & editing, Software, Validation, Investigation, Visualization. **Peerasait Prachaseree:** Conceptualization, Methodology, Formal analysis, Data curation, Writing – original draft, Writing – review & editing, Software, Validation, Investigation. **Emma Lejeune:** Conceptualization, Methodology, Formal analysis, Data curation, Writing – original draft, Writing – review & editing, Resources, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

All data to support this study is published through the OpenBU Institutional Repository.

Acknowledgments

We would like to thank the staff of the Boston University Research Computing Services and the OpenBU Institutional Repository (in particular Eleni Castro) for their invaluable assistance with generating and disseminating the datasets used in this paper. This work was made possible through start up funds from the Boston University Department of Mechanical Engineering, the David R. Dalton Career Development Professorship, the Hariri Institute Junior Faculty Fellowship, the Haythornthwaite Research Initiation Grant, the National Science Foundation, United States Grant CMMI-2127864, the American Heart Association Career Development Award 856354, and the Office of Naval Research, United States Grant N00014-22-1-2066.

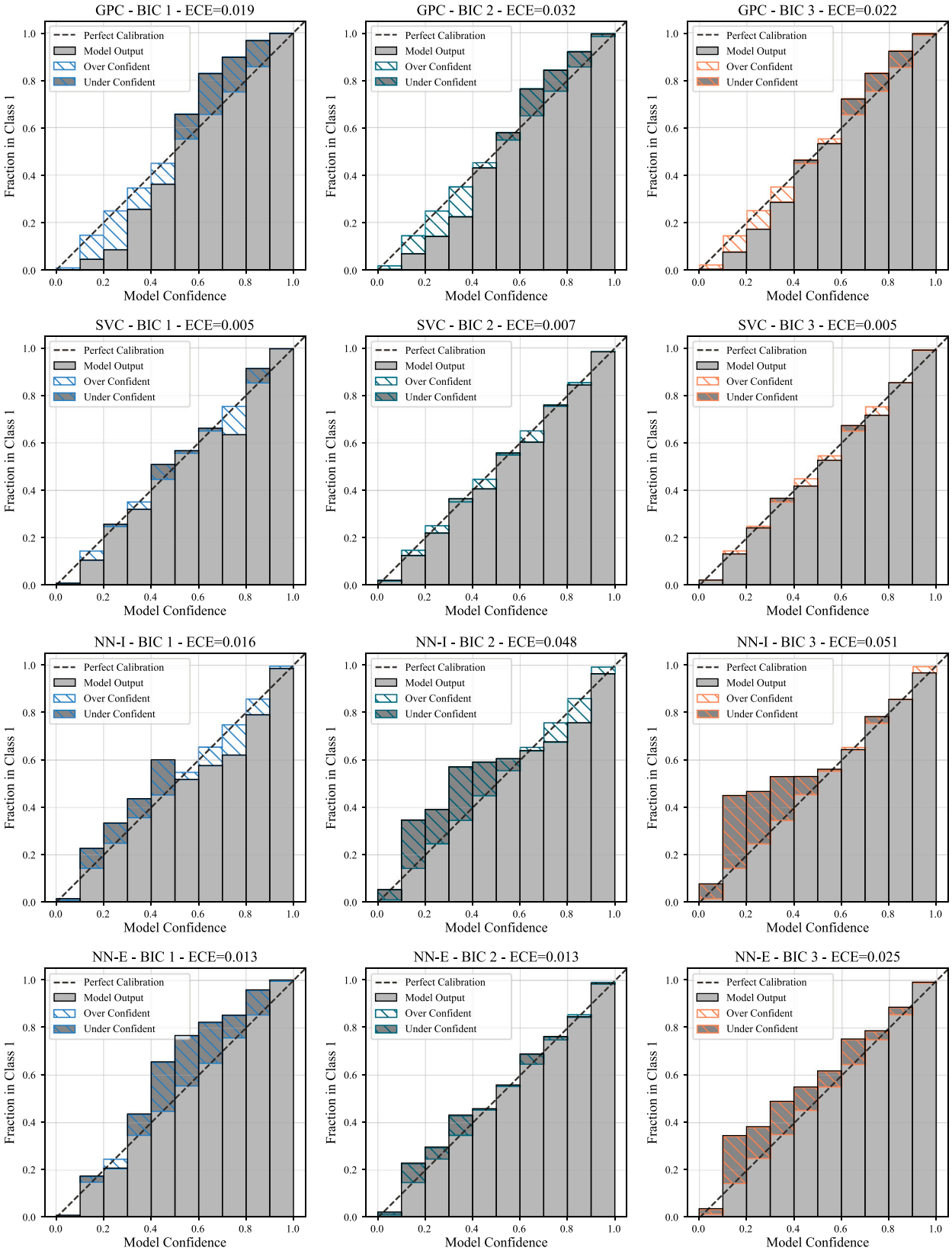

Fig. 7. Reliability diagrams as a supplement to Fig. 4.

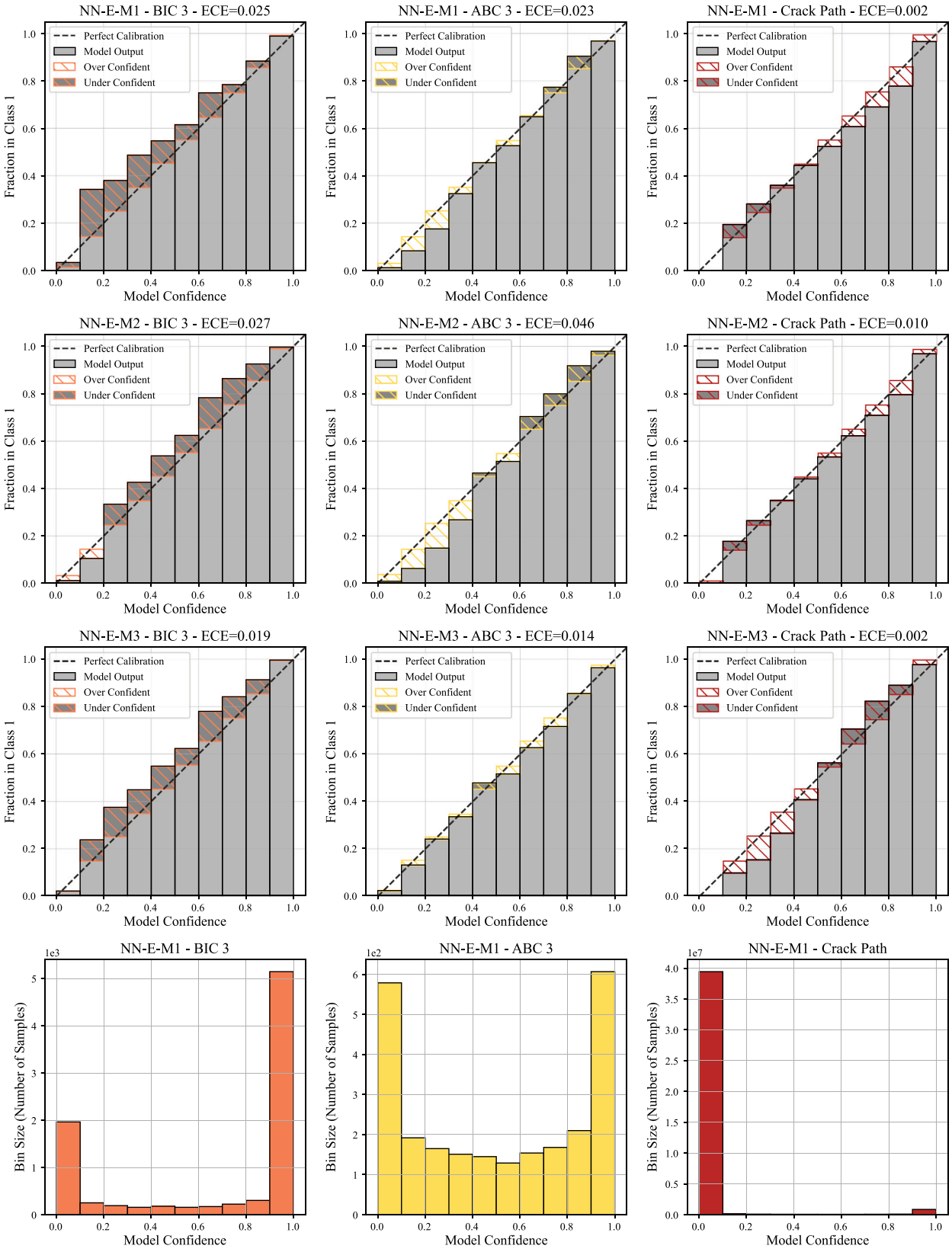

Fig. 8. Reliability diagrams and confidence distribution histograms as a supplement to Fig. 6.

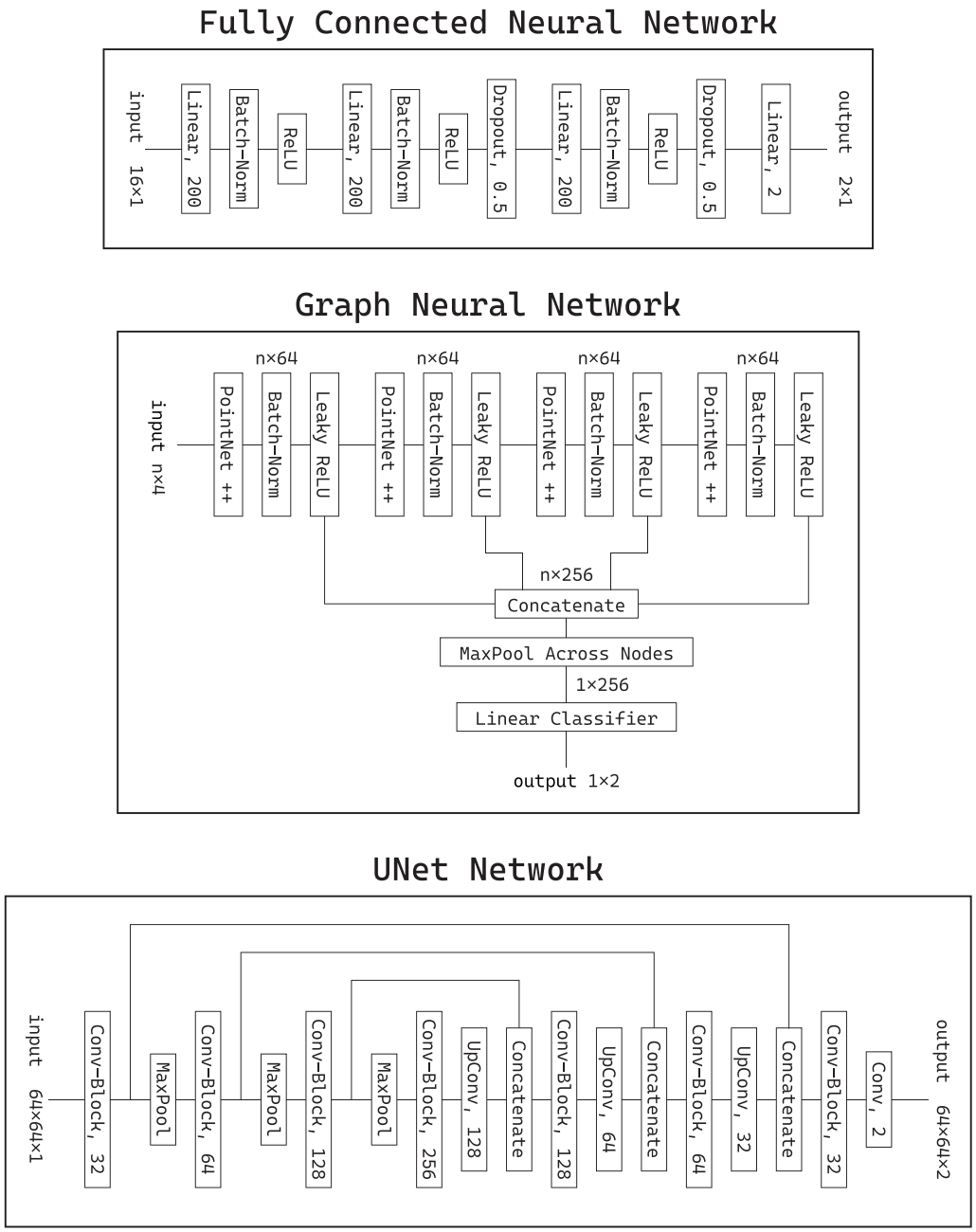

Fig. 9. This figure supplements Sections 2.2.3, 2.2.4, and 2.2.5 by schematically illustrating the different neural network architectures. The fully connected neural networks consists of three linear layers, each with 200 nodes, and utilizes dropout with a probability of 0.5 during training. Our previous work (Prachaseree and Lejeune, 2022b) provides further details on the graph neural network that we used. Finally, the UNet network employs "Conv-blocks", which repeat a 2D convolutional layer followed by a 2D batch normalization and a ReLU activation function twice. The convolutional layers have a filter size of 3 and a stride and padding of 1, while the maxpool layers use a filter size of 2. The transposed convolutional ("UpConv") layers employ a filter and stride of 2.

Appendix A. List of abbreviations

FEA - Finite Element Analysis
BIC - Buckling Instability Classification
ABC - Asymmetric Buckling Columns
NLL - Negative Log Likelihood
ECE - Expected Calibration Error
GPC - Gaussian Process Classification
SVC - Support Vector Classification
NN-I - Individual neural network (see Figs. 4 and 5)
NN-E - Ensemble neural network (see Figs. 4 and 5)

Method (I) - Individual neural network without post hoc calibration (see Fig. 6)
Method (I-C) - Individual neural network with post hoc calibration via temperature scaling (see Fig. 6)
Method (E-M1) - Ensemble neural network without post hoc calibration (see Fig. 6)
Method (E-M2) - Ensemble neural network with post hoc calibration via temperature scaling applied before ensemble averaging (see Fig. 6)
Method (E-M3) - Ensemble neural network with post hoc calibration via temperature scaling applied after en- semble averaging (see Fig. 6)

Appendix B. Supplementary reliability diagrams

In Section 3, we present the core results of this investigation as plots of Error vs. Expected Calibration Error (ECE). As introduced in Section 2.5.3 and illustrated in Fig. 3, ECE is connected to the reliability diagram, a common strategy for visualizing model calibration. Here we provide supplementary reliability diagrams to accompany Figs. 4 and 6. In Fig. 7, we show 12 calibration curves that correspond to the results shown in Fig. 4. Specifically, for BIC 1, BIC 2, and BIC 3 we plot reliability diagrams for Gaussian Process Classification, Support Vector Classification with Platt scaling, a representative Individual Neural Network, and an Ensemble Neural Network. In Fig. 8, we show 9 calibration curves that correspond to the results shown in Fig. 6. Specifically, for BIC 3, ABC 3, and Mechanical MNIST Crack Path we plot reliability diagrams for straightforward ensemble averaging (E-M1), post hoc calibration temperature scaling followed by ensemble averaging (E-M2), and ensemble averaging followed by post hoc calibration temperature scaling (E-M3). In addition, we plot histograms of the representative distribution of model confidence for E-M1 for BIC 3, ABC 3, and Mechanical MNIST Crack Path. These histograms not only indicate the bins that will have the heaviest weight, but also illustrate the severe class imbalance present in the Mechanical MNIST Crack Path dataset.

Beyond overall ECE, which is a weighted average of the calibration error in each bin, the reliability diagram allows us to visualize the Maximum Calibration Error and the regions where the different models tend to be over and under confident. In addition, they provide a visualization of the outcomes of post hoc model calibration via temperature scaling. These diagrams provide additional contextual information that helps address some of the limitations of ECE as a metric, discussed in Section 2.5.3. Overall, we recommend visualizing the reliability diagram in addition to computing ECE prior to deploying a given model.

Appendix C. Details of the neural networks

To supplement the description of the neural networks we used in this work, introduced in Sections 2.2.3, 2.2.4, and 2.2.5, we have included network schematics for our implementations of each network in Fig. 9.

References

Agarap, Abien Fred, 2018. Deep learning using rectified linear units (relu). arXiv preprint arXiv:1803.08375.

Alber, Mark, Buganza Tepole, Adrian, Cannon, William R., De, Suvranu, Dura-Bernal, Salvador, Garikipati, Krishna, Karniadakis, George, Lytton, William W., Perdikaris, Paris, Petzold, Linda, et al., 2019. Integrating machine learning and multiscale modeling—perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences. NPJ Digit. Med. 2 (1), 1–11.

Alnæs, Martin, Blechta, Jan, Hake, Johan, Johansson, August, Kehlet, Benjamin, Logg, Anders, Richardson, Chris, Ring, Johannes, Rognes, Marie E., Wells, Garth N., 2015. The FEniCS project version 1.5. Arch. Numer. Softw. 3 (100).

Amini, Alexander, Schwarting, Wilko, Soleimany, Ava, Rus, Daniela, 2020. Deep evidential regression. Adv. Neural Inf. Process. Syst. 33, 14927–14937.

Ardizzone, Lynton, Kruse, Jakob, Wirkert, Sebastian, Rahner, Daniel, Pellegrini, Eric W., Klessen, Ralf S., Maier-Hein, Lena, Rother, Carsten, Köthe, Ullrich, 2018. Analyzing inverse problems with invertible neural networks. arXiv preprint arXiv:1808.04730.

Arendt, Paul D., Apley, Daniel W., Chen, Wei, 2012. Quantification of model uncertainty: Calibration, model discrepancy, and identifiability.

Bartók, Albert P., Kermode, James, et al., 2022. Improved uncertainty quantification for Gaussian process regression based interatomic potentials. arXiv preprint arXiv:2206.08744.

Ciregan, Dan, Meier, Ueli, Schmidhuber, Jürgen, 2012. Multi-column deep neural networks for image classification. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 3642–3649.

Deng, Jia, Dong, Wei, Socher, Richard, Li, Li-Jia, Li, Kai, Fei-Fei, Li, 2009. Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. Ieee, pp. 248–255.

Do, Hung P., Guo, Yi, Yoon, Andrew J., Nayak, Krishna S., 2020. Accuracy, uncertainty, and adaptability of automatic myocardial ASL segmentation using deep CNN. Magn. Reson. Med. 83 (5), 1863–1874.

Drucker, Harris, Burges, Christopher J., Kaufman, Linda, Smola, Alex, Vapnik, Vladimir, 1996. Support vector regression machines. Adv. Neural Inf. Process. Syst. 9.

Duvenaud, David, 2014. The kernel cookbook: Advice on covariance functions. URL https://www.cs.toronto.edu/~duvenaud/cookbook.

Elhassouny, Azeddine, Smarandache, Florentin, 2019. Trends in deep convolutional neural Networks architectures: A review. In: 2019 International Conference of Computer Science and Renewable Energies. ICCSRE, IEEE, pp. 1–8.

Fey, Matthias, Lenssen, Jan E., 2019. Fast graph representation learning with PyTorch Geometric. In: ICLR Workshop on Representation Learning on Graphs and Manifolds.

Gander, Lia, Pezzuto, Simone, Gharaviri, Ali, Krause, Rolf, Perdikaris, Paris, Sahli Costabal, Francisco, 2022. Fast characterization of inducible regions of atrial fibrillation models with multi-fidelity Gaussian process classification. Front. Physiol. 260.

Geuzaine, Christophe, Remacle, Jean-François, 2009. Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities. Internat. J. Numer. Methods Engrg. 79 (11), 1309–1331.

Gneiting, Tilmann, Raftery, Adrian E., 2007. Strictly proper scoring rules, prediction, and estimation. J. Amer. Statist. Assoc. 102 (477), 359–378.

Gongora, Aldair E., Snapp, Kelsey L., Pang, Richard, Tiano, Thomas M., Reyes, Kristofer G., Whiting, Emily, Lawton, Timothy J., Morgan, Elise F., Brown, Keith A., 2022. Designing lattices for impact protection using transfer learning. Matter 5 (9), 2829–2846.

Guo, Chuan, Pleiss, Geoff, Sun, Yu, Weinberger, Kilian Q., 2017. On calibration of modern neural networks. In: International Conference on Machine Learning. PMLR, pp. 1321–1330.

Guo, Kai, Yang, Zhenze, Yu, Chi-Hua, Buehler, Markus J., 2021. Artificial intelligence and machine learning in design of mechanical materials. Mater. Horiz. 8 (4), 1153–1172.

Han, Tianhong, Ahmed, Kaleem S., Gosain, Arun K., Tepole, Adrian Buganza, Lee, Taeksang, 2022. Multi-fidelity Gaussian process surrogate modeling of pediatric tissue expansion. J. Biomech. Eng. 144 (12), 121005.

Hanakata, Paul Z., Cubuk, Ekin D., Campbell, David K., Park, Harold S., 2020. Forward and inverse design of kirigami via supervised autoencoder. Phys. Rev. Res. 2 (4), 042006.

Hearst, Marti A., Dumais, Susan T., Osuna, Edgar, Platt, John, Scholkopf, Bernhard, 1998. Support vector machines. IEEE Intell. Syst. Appl. 13 (4), 18–28.

Ioffe, Sergey, Szegedy, Christian, 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning. PMLR, pp. 448–456.

Jadon, Shruti, 2020. A survey of loss functions for semantic segmentation. In: 2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology. CIBCB, IEEE, pp. 1–7.

Jin, Zeqing, Zhang, Zhizhou, Gu, Grace X., 2020. Automated real-time detection and prediction of interlayer imperfections in additive manufacturing processes using artificial intelligence. Adv. Intell. Syst. 2 (1), 1900130.

Joshi, Akshay, Thakolkaran, Prakash, Zheng, Yiwen, Escande, Maxime, Flaschel, Moritz, De Lorenzis, Laura, Kumar, Siddhant, 2022. Bayesian-EUCLID: Discovering hyperelastic material laws with uncertainties. Comput. Methods Appl. Mech. Engrg. 398, 115225.

Kapteyn, Michael G., Pretorius, Jacob V.R., Willcox, Karen E., 2021. A probabilistic graphical model foundation for enabling predictive digital twins at scale. Nat. Comput. Sci. 1 (5), 337–347.

Karapiperis, K., Stainier, L., Ortiz, M., Andrade, J.E., 2021. Data-driven multiscale modeling in mechanics. J. Mech. Phys. Solids 147, 104239.

Kendall, Alex, Gal, Yarin, 2017. What uncertainties do we need in bayesian deep learning for computer vision? Adv. Neural Inf. Process. Syst. 30.

Kingma, Diederik P., Ba, Jimmy, 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kissas, Georgios, Seidman, Jacob H., Guilhoto, Leonardo Ferreira, Preciado, Victor M., Pappas, George J., Perdikaris, Paris, 2022. Learning operators with coupled attention. J. Mach. Learn. Res. 23 (215), 1–63.

Kuleshov, Volodymyr, Fenner, Nathan, Ermon, Stefano, 2018. Accurate uncertainties for deep learning using calibrated regression. In: International Conference on Machine Learning. PMLR, pp. 2796–2804.

Lakshminarayanan, Balaji, Pritzel, Alexander, Blundell, Charles, 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. Adv. Neural Inf. Process. Syst. 30.

LeCun, Yann, Cortes, Corinna, 2010. MNIST handwritten digit database.

Lejeune, Emma, 2020a. Buckling instability classification (BIC). https://open.bu.edu/handle/2144/40085.

Lejeune, Emma, 2020b. Mechanical MNIST: A benchmark dataset for mechanical metamodels. Extreme Mech. Lett. 36, 100659.

Lejeune, Emma, 2021. Geometric stability classification: datasets, metamodels, and adversarial attacks. Comput. Aided Des. 131, 102948.

Logg, Anders, Mardal, Kent-Andre, Wells, Garth, 2012. Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book, Vol. 84. Springer Science & Business Media.

Maddox, Wesley J., Izmailov, Pavel, Garipov, Timur, Vetrov, Dmitry P., Wilson, Andrew Gordon, 2019. A simple baseline for bayesian uncertainty in deep learning. Adv. Neural Inf. Process. Syst. 32.

Mann, Andrew, Kalidindi, Surya R., 2022. Development of a robust CNN model for capturing microstructure-property linkages and building property closures supporting material design. Virtual Mater. Des. 879614107.

Mehrtash, Alireza, Wells, William M., Tempany, Clare M., Abolmaesumi, Purang, Kapur, Tina, 2020. Confidence calibration and predictive uncertainty estimation for deep medical image segmentation. IEEE Trans. Med. Imaging 39 (12), 3868–3878.

Minderer, Matthias, Djolonga, Josip, Romijnders, Rob, Hubis, Frances, Zhai, Xiaohua, Houlsby, Neil, Tran, Dustin, Lucic, Mario, 2021. Revisiting the calibration of modern neural networks. Adv. Neural Inf. Process. Syst. 34, 15682–15694.

Mohammadzadeh, Saeed, Lejeune, Emma, 2021. Mechanical MNIST crack path. https://open.bu.edu/handle/2144/42757.

Mohammadzadeh, Saeed, Lejeune, Emma, 2022. Predicting mechanically driven full-field quantities of interest with deep learning-based metamodels. Extreme Mech. Lett. 50, 101566.

Naeini, Mahdi Pakdaman, Cooper, Gregory, Hauskrecht, Milos, 2015. Obtaining well calibrated probabilities using bayesian binning. In: Twenty-Ninth AAAI Conference on Artificial Intelligence.

Niculescu-Mizil, Alexandru, Caruana, Rich, 2005. Predicting good probabilities with supervised learning. In: Proceedings of the 22nd International Conference on Machine Learning. pp. 625–632.

Nixon, Jeremy, Dusenberry, Michael W., Zhang, Linchuan, Jerfel, Ghassen, Tran, Dustin, 2019. Measuring calibration in deep learning. In: CVPR Workshops, Vol. 2.

Ovadia, Yaniv, Fertig, Emily, Ren, Jie, Nado, Zachary, Sculley, David, Nowozin, Sebastian, Dillon, Joshua, Lakshminarayanan, Balaji, Snoek, Jasper, 2019. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. Adv. Neural Inf. Process. Syst. 32.

Paszke, Adam, Gross, Sam, Massa, Francisco, Lerer, Adam, Bradbury, James, Chanan, Gregory, Killeen, Trevor, Lin, Zeming, Gimelshein, Natalia, Antiga, Luca, et al., 2019. Pytorch: An imperative style, high-performance deep learning library. Adv. Neural Inf. Process. Syst. 32.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in Python. J. Mach. Learn. Res. 12, 2825–2830.

Perdikaris, Paris, Venturi, Daniele, Royset, Johannes O., Karniadakis, George Em, 2015. Multi-fidelity modelling via recursive co-kriging and Gaussian–Markov random fields. Proc. R. Soc. A: Math. Phys. Eng. Sci. 471 (2179), 20150018.

Platt, John, et al., 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: Advances in Large Margin Classifiers, Vol. 10. pp. 61–74, (3).

Prachaseree, Peerasait, Lejeune, Emma, 2022a. Asymmetric buckling columns (ABC). https://open.bu.edu/handle/2144/43730.

Prachaseree, Peerasait, Lejeune, Emma, 2022b. Learning mechanically driven emergent behavior with message passing neural networks. Comput. Struct. 270, 106825.

Psaros, Apostolos F., Meng, Xuhui, Zou, Zongren, Guo, Ling, Karniadakis, George Em, 2022. Uncertainty quantification in scientific machine learning: Methods, metrics, and comparisons. arXiv preprint arXiv:2201.07766.

Qi, Charles R., Yi, Li, Su, Hao, Guibas, Leonidas J., 2017. PointNet++: Deep hierarchical feature learning on point sets in a metric space. arXiv preprint arXiv:1706.02413.

Rahaman, Rahul, Thiery, Alexandre H., 2020. Uncertainty quantification and deep ensembles. arXiv preprint arXiv:2007.08792.

Rahimi, Amir, Gupta, Kartik, Ajanthan, Thalaiyasingam, Mensink, Thomas, Sminchisescu, Cristian, Hartley, Richard, 2020. Post-hoc calibration of neural networks. arXiv preprint arXiv:2006.12807.

Raissi, Maziar, Perdikaris, Paris, Karniadakis, George E., 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. 378, 686–707.

Ronneberger, Olaf, Fischer, Philipp, Brox, Thomas, 2015. U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 234–241.

Sensoy, Murat, Kaplan, Lance, Kandemir, Melih, 2018. Evidential deep learning to quantify classification uncertainty. Adv. Neural Inf. Process. Syst. 31.

Shalev-Shwartz, Shai, Ben-David, Shai, 2014. Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press.

Shin, Dongil, Cupertino, Andrea, de Jong, Matthijs H.J., Steeneken, Peter G., Bessa, Miguel A., Norte, Richard A., 2022. Spiderweb nanomechanical resonators via bayesian optimization: inspired by nature and guided by machine learning. Adv. Mater. 34 (3), 2106248.

Siddique, Nahian, Paheding, Sidike, Elkin, Colin P., Devabhaktuni, Vijay, 2021. U-net and its variants for medical image segmentation: A review of theory and applications. Ieee Access 9, 82031–82057.

Srivastava, Nitish, Hinton, Geoffrey, Krizhevsky, Alex, Sutskever, Ilya, Salakhutdinov, Ruslan, 2014. Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15 (1), 1929–1958.

Sun, Xiao, Bahmani, Bahador, Vlassis, Nikolaos N., Sun, WaiChing, Xu, Yanxun, 2022. Data-driven discovery of interpretable causal relations for deep learning material laws with uncertainty propagation. Granul. Matter 24 (1), 1–32.

Vlassis, Nikolaos N., Ma, Ran, Sun, WaiChing, 2020. Geometric deep learning for computational mechanics Part I: anisotropic hyperelasticity. Comput. Methods Appl. Mech. Engrg. 371, 113299.

Wang, Liwei, Chan, Yu-Chin, Ahmed, Faez, Liu, Zhao, Zhu, Ping, Chen, Wei, 2020a. Deep generative modeling for mechanistic-based learning and design of metamaterial systems. Comput. Methods Appl. Mech. Engrg. 372, 113377.

Wang, Zhenlin, Huan, Xun, Garikipati, Krishna, 2019. Variational system identification of the partial differential equations governing the physics of pattern-formation: Inference under varying fidelity and noise. Comput. Methods Appl. Mech. Engrg. 356, 44–74.

Wang, Zhenlin, Wu, Bowei, Garikipati, Krishna, Huan, Xun, 2020b. A perspective on regression and Bayesian approaches for system identification of pattern formation dynamics. Theor. Appl. Mech. Lett. 10 (3), 188–194.

Weston, Jason, Watkins, Chris, 1998. Multi-Class Support Vector Machines. Technical report, Citeseer.

Williams, Christopher K.I., Rasmussen, Carl Edward, 2006. Gaussian Processes for Machine Learning, Vol. 2. MIT press, Cambridge, MA.

Wu, Jian-Ying, 2017. A unified phase-field theory for the mechanics of damage and quasi-brittle failure. J. Mech. Phys. Solids 103, 72–99.

Wu, Jian-Ying, Nguyen, Vinh Phu, Nguyen, Chi Thanh, Sutula, Danas, Sinaie, Sina, Bordas, Stéphane P.A., 2020. Phase-field modeling of fracture. Adv. Appl. Mech. 53, 1–183.

Xiao, Han, Rasul, Kashif, Vollgraf, Roland, 2017. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms.

Yang, Yibo, Perdikaris, Paris, 2019. Adversarial uncertainty quantification in physics-informed neural networks. J. Comput. Phys. 394, 136–152.

Yin, Minglang, Zhang, Enrui, Yu, Yue, Karniadakis, George Em, 2022. Interfacing finite elements with deep neural operators for fast multiscale modeling of mechanics problems. Comput. Methods Appl. Mech. Engrg. 402, 115027.

Zadrozny, Bianca, Elkan, Charles, 2002. Transforming classifier scores into accurate multiclass probability estimates. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 694–699.

Zandigohar, Mehrshad, Erdoğmuş, Deniz, Schirner, Gunar, 2021. Netcut: Real-time dnn inference using layer removal. In: 2021 Design, Automation & Test in Europe Conference & Exhibition. DATE, IEEE, pp. 1845–1850.

Zhang, Xiaoxuan, Garikipati, Krishna, 2021. Bayesian neural networks for weak solution of pdes with uncertainty quantification. arXiv preprint arXiv:2101.04879.

Zhang, Jize, Kailkhura, Bhavya, Han, T. Yong-Jin, 2020. Mix-n-match: Ensemble and compositional methods for uncertainty calibration in deep learning. In: International Conference on Machine Learning. PMLR, pp. 11117–11128.