Pyle, R., Hughes, R. R., Ait Si Ali, A., & Wilcox, P. D. (2022). Uncertainty Quantification for Deep Learning in Ultrasonic Crack Characterization. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, *69*(7), 2339 - 2351. https://doi.org/10.1109/TUFFC.2022.3176926

Uncertainty Quantification for Deep Learning in Ultrasonic Crack Characterization

Richard J. Pyle, Robert R. Hughes, Amine Ait Si Ali, Paul D. Wilcox, *Member, IEEE*

***Abstract*—Deep learning for Non-Destructive Evaluation (NDE) has received a lot of attention in recent years for its potential ability to provide human level data analysis. However, little research into quantifying the uncertainty of its predictions has been done. Uncertainty Quantification (UQ) is essential for qualifying NDE inspections and building trust in their predictions. Therefore, this paper aims to demonstrate how UQ can best be achieved for deep learning in the context of crack sizing for inline pipe inspection. A convolutional neural network architecture is used to size surface breaking defects from Plane Wave Imaging (PWI) images with two modern UQ methods: deep ensembles and Monte Carlo dropout. The network is trained using PWI images of surface breaking defects simulated with a hybrid finite element / ray-based model.**

Successful UQ is judged by calibration and anomaly detection, which refer to whether in-domain model error is proportional to uncertainty and if out of training domain data is assigned high uncertainty, respectively. Calibration is tested using simulated and experimental images of surface breaking cracks, while anomaly detection is tested using experimental side drilled holes and simulated embedded cracks. Monte Carlo dropout demonstrates poor uncertainty quantification with little separation between in and out-of-distribution data and a weak linear fit ($R = 0.84$) between experimental root mean squared error and uncertainty. Deep ensembles improve upon Monte Carlo dropout in both calibration ($R = 0.95$) and anomaly detection. Adding spectral normalization and residual connections to deep ensembles slightly improves calibration ($R = 0.98$) and significantly improves the reliability of assigning high uncertainty to out-of-distribution samples.

***Index Terms*— Uncertainty estimation, out-of-distribution detection, deep-learning, neural networks, plane wave imaging, simulation, ultrasound, defect characterization**

I. Introduction

NON-Destructive Evaluation (NDE) techniques aim to infer the health of a component through analysis of its response to a stimulus such as ultrasound or X-ray. In most NDE applications this is conventionally achieved by a skilled operator inspecting the response data. As this data is often high-dimensional, and most inspections must be carried out many times, manual data interpretation is expensive and prone to human error. Because of this, there is a strong case for automating data interpretation in NDE. Machine learning is well suited to pattern recognition tasks such as this one and has repeatedly been shown to produce human-level data interpretation performance both in NDE [1]–[9], and related fields such as computer vision [10] and medical imaging [11]. In safety critical applications such as NDE it is essential to know the magnitude of expected error so reports on a component's health can be given with an appropriate level of confidence. Calculating the level of expected error for a prediction is commonly called Uncertainty Quantification (UQ). Despite the acute need for UQ in NDE there has been little research into how to implement it for analysis done using machine learning.

Machine learning can broadly be split into two categories: 'deep' and 'shallow' learning. Shallow learning necessitates hand-selection of the features input to the machine learning algorithm, while deep learning takes the raw data as input. Shallow learning is a lower dimensional problem so requires less training data, and because manufacturing NDE samples is expensive, it has been the focus of most NDE research to-date. However, deep learning can make use of all available information in the data so given sufficient training data it can produce more accurate results [4], [5] and reduce the effect of human factors [12]. Deep learning is the focus of this paper. To produce the training set, data simulation is used as it has recently been shown to be an effective way of training Convolutional Neural Networks (CNNs) [13] to accurately size defects in experimental data [9], [14]–[17].

Due to the safety-critical nature of NDE, UQ is an essential part of inspection qualification [18] and decision making for any automated data analysis. This is because undersizing of defects can result in unexpected part failures, causing damage to structures and/or people. Effective UQ can signal to the operator when there is high uncertainty in the defect size prediction so the data can be referred to a human for further analysis and possibly the use of additional NDE measurements. This paper focusses on how to quantify uncertainty for deep learning in the context of crack sizing in ultrasonic inline pipe inspection. Ultrasonic inline pipe inspection uses transducers mounted on a PIG (Pipeline Inspection Gauge) which travels in the flow of product, detecting and size defects in the surrounding pipe wall. Automatic defect detection occurs

Submitted for publication on 16/05/22. This research is funded by the Engineering and Physical Sciences Research Council (EPSRC, grant number EP/L015587/1) via the Research Centre for Non-Destructive Evaluation (RCNDE), with additional funding provided by Baker Hughes, Cramlington, UK.

Richard Pyle, Robert Hughes and Paul Wilcox are with the Department of Mechanical Engineering, University of Bristol, UK

Richard Pyle and Amine Ait Si Ali are with Baker Hughes, Cramlington, UK

online, and in this paper is assumed to have already been performed, hence the task is to characterize and size a defect given data that contains an indication of a defect. Defect sizing occurs offline and is traditionally carried out by skilled human operators. In this paper deep learning is applied to the defect characterisation and sizing task with the aim of investigating how the uncertainty of that operation can be assessed

Evaluating the success of UQ methods is challenging as there is no 'ground-truth' for uncertainty. This paper uses two criteria to analyse the success of the UQ methods. The first is for the UQ method to be 'well calibrated' [19]. For regression tasks, such as the one in this paper, this means that predicted uncertainty is equal to (or at least proportional to) the expected error (i.e. the difference between the crack depth predicted by the network and the true crack depth). This is tested using both a simulated and experimental test set of surface breaking cracks. The second metric is the predicted uncertainty for Out-Of-Distribution (OOD) data, testing if the network 'knows what it knows.' As the network is trained on surface-breaking cracks, OOD data from experimental embedded Side-Drilled Holes (SDHs) and simulated embedded cracks are used for this purpose. The OOD data set ($N_{OOD} = 76$) contains examples of defects not included in the training data and therefore an effective UQ method should assign them high uncertainty

In practice, as in this paper, UQ typically produces a single metric, e.g. standard deviation of the probability density function, $P(\hat{y}|\hat{x}, D)$, where $\hat{x}$, $\hat{y}$ are the network's input and output for test data and D is the input and output training data. The methods described in this paper achieve UQ by sampling from the space of all possible trained networks (parameterized by their weights, W) and taking the standard deviation of their predictions as an estimate of uncertainty. In more rigorous terms, all UQ methods function by approximating the intractable posterior distribution of weights given the labelled training data, $P(W|D)$, with which inference on the uncertainty associated with new test data, $P(\hat{y}|\hat{x}, D)$, can be calculated. The two most common modern methods for estimating the uncertainty of the CNN's predictions are investigated for this paper: deep ensembles (DE) [20] and Monte Carlo (MC) dropout [21]. The intuition for these approaches to posterior approximation is that if the sampled networks are sufficiently diverse, they should produce diverse predictions for inputs far from the training data, indicating high uncertainty. DE achieves this by training multiple networks from different initializations, while MC dropout produces predictions by using dropout (traditionally used at train time to reduce overfitting [22]) at test time.

The structure of the rest of the paper is as follows. Relevant literature is discussed in Section II, the inspection setup, data-sets and network architecture are described in Section III, the two UQ methods presented in this paper are outlined in Section IV, results are presented in Section V, methods for efficient use of computational resources are discussed in Section VI and conclusions are given in Section VII.

II. Relevant Literature

UQ is a relatively new and active area of research in deep learning [23]. Because of this, there are few applications to NDE in the literature. To the authors' knowledge the only examples of UQ for deep learning in NDE are the following: MC dropout used to estimate uncertainty for defect detection in a heat exchanger with eddy-current measurements [24] as well as for defect categorization and localization in visual inspection of bridges [25]. A mixture density network [26] has been used to estimate aleatoric uncertainty for guided wave based defect localization in simulated data of structural plates [27]. Deep ensembles have been used to increase the accuracy of deep learnt predictions in NDE [28]–[30], but there has been little investigation into leveraging their ability to quantify uncertainty.

While this paper focusses primarily on DE and MC dropout, two other commonly used UQ methods were investigated in the formation of this paper: a CNN/Gaussian Process (CNN-GP) hybrid [31], [32], and Variational Inference (VI) [33], [34]. These methods take a more 'Bayesian' rather than 'Frequentist' approach to approximation of the posterior. CNN-GP makes use of the natural probabilistic inference of the Gaussian process combined with the expressive powers of convolutional layers. Following the implementation described in [32], the fully connected layers of a CNN were replaced with a sparse Gaussian process approximation based on variational inducing points [35] for the current application. This method was found to produce no correlation between uncertainty and magnitude of error on the experimental test set. VI approximates the posterior by casting it as an optimization problem: reducing the Kullback-Leibler divergence [36] between the true posterior and that produced by the network. For the application described in the current paper, VI was implemented using a reparameterization estimator [37]. However, VI proved to be unstable in training and converged either to a network predicting the mean of the training set or one with poor predictive accuracy (sizing defects with a root mean square error ≈ 0.4 times their true length). There have also been recent publications that question the quality of VI's posterior approximation [45]–[47]. As these methods require a lot of hyperparameter tuning and, despite this, were found to produce poor UQ, they are not investigated further in this paper.

III. Inspection Setup, Data and Network Architecture

This section describes the inspection set-up as well as the model used to simulate PWI data, experimental and OOD data sets, and the details of the CNN architecture. The reader is directed to [9] for more details. Note that for clarity, 'model' is used exclusively to describe physics-based forward models while 'network' is used to refer to machine learning based predictors.

A. Inspection Setup, Imaging and Simulation

Inline pipe inspection methods are typically used to inspect oil and gas pipelines. A major aim of these inspections is to

detect cracks in the pipe caused by manufacturing faults or in-service stresses. While these can occur at any radial location this work focusses on surface breaking cracks on the outer surface as this is the most common location for them to occur. With access to a real pipeline not available for this work an inspection setup is devised to match in-service conditions as closely as possible. As shown in Fig. 1a, an Imasonic (Voray-sur-l'Ognon, France) 5MHz, 0.3 mm pitch, 40 element phased array is used to induce shear plane waves in 10 mm thick stainless-steel plate (approximating a large diameter pipe wall). The array is operated using a Peak NDT (Derby, UK) MicroPulse 5 array controller and receives on all elements individually, with a sample rate of 50MHz, to form Plane Wave Capture (PWC) data. The array is immersed in water as an approximation for oil that has similar sound speed.

As shown in Fig. 1b, data is collected from either side of the defect to replicate the use of a pair of arrays from the circumferential ring of arrays on the PIG. Each of these arrays fires a vertical wave at $\psi = 0°$ and an angled wave that travels in the fluid at $\phi = \pm 19°$, inducing a $\psi = \pm 45°$ shear wave in the steel plate. The vertical wave is used to calculate standoff (ς) and thickness (Γ), while all sizing is done using the angled waves. The arrays receive on all 40 elements individually to collect PWC data which is then filtered using a Gaussian filter centered at 5 MHz with a -40 dB half width of 4.5 MHz. This filtered PWC data is then focused on reception with the overall process termed Plane Wave Imaging (PWI) [38]. When multiple ray paths are considered, the images are termed 'views', and are described by the modality(s) of their transmit and receive legs (L for longitudinal, S for shear) separated by a hyphen to indicate reflection from a defect. The two views found to be most successful for sizing the surface breaking defects used in this paper are the SS-S and SS-L half-skip views (i.e. with a reflection off the back wall of the plate on the transmit leg only). Each array produces an SS-S and SS-L image for each defect, with the region of interest being the full 10 mm depth of plate thickness and 12-22 mm from the array centre in the X-direction. This results in a 32x32x4 set of data as input to the network. Example sets of simulated and experimental images for a defect of $P = 19$ mm, $L = 3$ mm and $\theta = 8°$ is given in Fig. 1c,d.

The simulation used to create the training data for this paper is a hybrid Finite Element (FE)/ray-based method which provides a good tradeoff between computational efficiency and accuracy. This simulation functions by calculating the scattering matrix for all length (L) and angle (θ) combinations

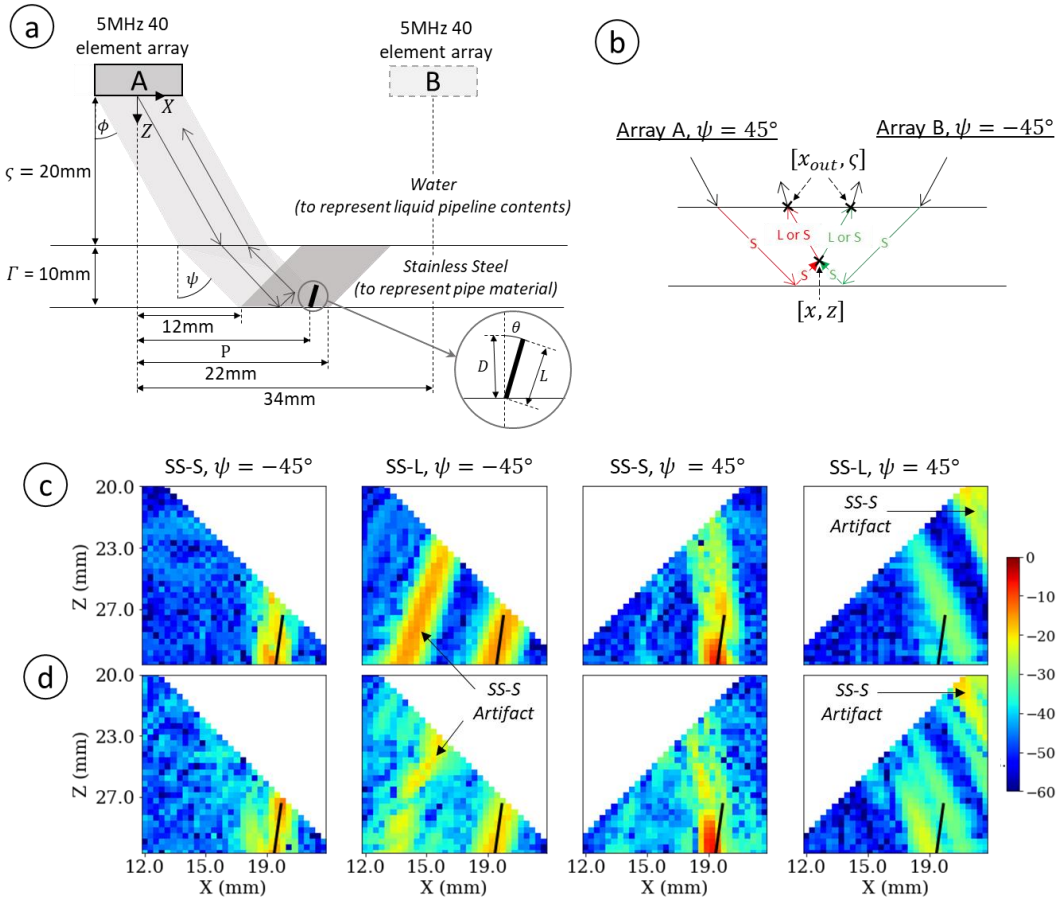

Fig. 1. a) A diagram of the inspection scenario using a plane wave at angle ψ to the vertical transmitted in the sample with a standoff and thickness of ς and Γ where L, θ and P represent the crack length, angle and position respectively, b) all half-skip shear (S) and longitudinal (L) mode ray-paths used in this paper where x, z are the co-ordinates of the imaging point and x_{out}, ς the co-ordinates of the returning ray on the front wall, c) an example set of simulated images for a defect with $P = 19\ mm, L = 3\ mm$ and $\theta = 8°$ and d) a fully experimental set of images for a defect of the same parameters. Note that the black lines show the true extent of the defects and all images are on the same dB color scale, normalized to the maximum intensity in the experimental set. Figure reproduced from [9].

using a local FE calculation, where the defect is modelled as a 0.3 mm wide rectangular, perfect reflector, excited by a unimodal plane wave [39]. This scattering matrix can then be used in a ray-based model [40], [41] to calculate the PWC data received by the array for every L, θ and position (P) combination before being summed with experimental, defect-free PWC data to incorporate grain and wall reflections [42]. Finally, the PWC data is filtered and imaged in the same way as the experimental data to produce the desired PWI images.

B. Data Sets

This paper focusses mainly on quantifying uncertainty for sizing surface-breaking cracks but data from other defects is also tested to analyse the predicted uncertainty for OOD defects. All of the data used in this paper and their main sources of uncertainty are described in this section.

1) Surface Breaking Cracks

The simulation and experimental procedures described in the previous section are used to generate data sets of size 16,875 and 1,485 respectively which are both further broken down into the following sets:

Simulated, training: 85% (14,343) of simulated data used to iteratively update the weights and biases of the network.

Simulated, validation: 7.5% (1,266) of simulated data used during research and design stages to qualitatively ensure the network is not overfitting to the training set.

Simulated, testing: 7.5% (1,266) of simulated data used to test the calibration of UQ on previously unseen data.

Experimental, validation: 15% (216) of experimental data used during research and design stages to ensure the network is not overfitting to the simulated data and to implement the training stop condition.

Experimental, testing: 85% (1,269) of experimental data used to test the network's sizing accuracy and calibration of UQ on previously unseen data.

These data sets are described further in Tables I and II. The training/validation/testing split for simulated data is drawn randomly, from a uniform distribution, across all image sets (i.e. across all $\{L, \theta, P\}$), but the experimental validation/testing split is drawn randomly in $\{L, \theta\}$ space. This is to guarantee that no data from the same physical defect is split across sets, ensuring test set performance generalizes past the L, θ combinations used to implement the stop condition. The aim of these surface breaking defect test sets is to analyse the calibration between uncertainty and D prediction error.

2) Defects Outside of Training Set

To test whether the UQ methods can detect data drawn from distributions significantly different to the training set, defect types not included in the training set are tested. As exampled in Fig. 2, this group of data includes two experimental Side Drilled Holes (SDHs) and two simulated embedded (rather than surface-breaking) cracks. This data is gathered using the same experimental and simulation procedures as described in Section III.A. These four defect classes are imaged at 14 X-locations, equally spaced across the same range of horizontal positions as the surface breaking cracks (13 mm $\leq P \leq$ 21 mm). 20 examples of experimental defect free data are also tested, forming a total of $N_{OOD} = 4 \times 14 + 20 = 76$ image sets.

TABLE I
Simulated Data Set Summary

Parameter	*Range*	*Step*	*Count*
Crack Length, L (mm)	0.2 to 5	0.2	25
Crack Position, P (mm)	13 to 21	0.3	27
Crack Angle, θ (°)	-24 to 24	2	25
Non-Defect Scan	-	-	36
Total = 25×27×25 = **16,875 image sets**			

TABLE II
Experimental Data Set Summary.
The experimental test set contains all of the L/θ combinations marked 'Test' while the experimental validation set all those marked 'Val.'

		Crack Length, L (mm)				
		1	2	3	4	5
Crack Angle, θ (°)	0	Test	Test	Test	Test	Test
	±2	Test	**Val**	Test	Test	Test
	±5	**Val**	Test	Test	Test	Test
	±8	Test	Test	Test	**Val**	Test
	±15	Test	Test	Test	Test	Test
	±20	Test	Test	**Val**	Test	Test

	Range	*Step*	*Count*
Crack Position, P (mm)	13 to 21	0.3	27

Validation = $N_{\theta,L} \times N_P$ = 8 × 27 = **216 image sets**

Test = $N_{\theta,L} \times N_P$ = 47 × 27 = **1269 image sets**

3) Sources of Uncertainty

Sources of uncertainty can broadly be broken down into two categories; aleatoric and epistemic. Aleatoric or 'data' uncertainty stems from noise inherent to the data generation process, and cannot be reduced by adding training data. Epistemic uncertainty is caused by ignorance in how the data is generated, creating uncertainty in the network's parameters, and can be minimized by adding appropriate training data as long as the training data chosen matches the test data distribution well. It should be highlighted that if there is a significant domain shift between training and test domains (e.g. when using a numerical simulation to approximate reality) adding training data can never fully minimize epistemic uncertainty.

In sizing defects from PWI images the two main sources of aleatoric uncertainty are noise and poor correlation between indication and defect size. Noise is caused by reflections from grains and structural features (such as front and back walls), as well as "artifacts" at locations away from the defect, due to ray

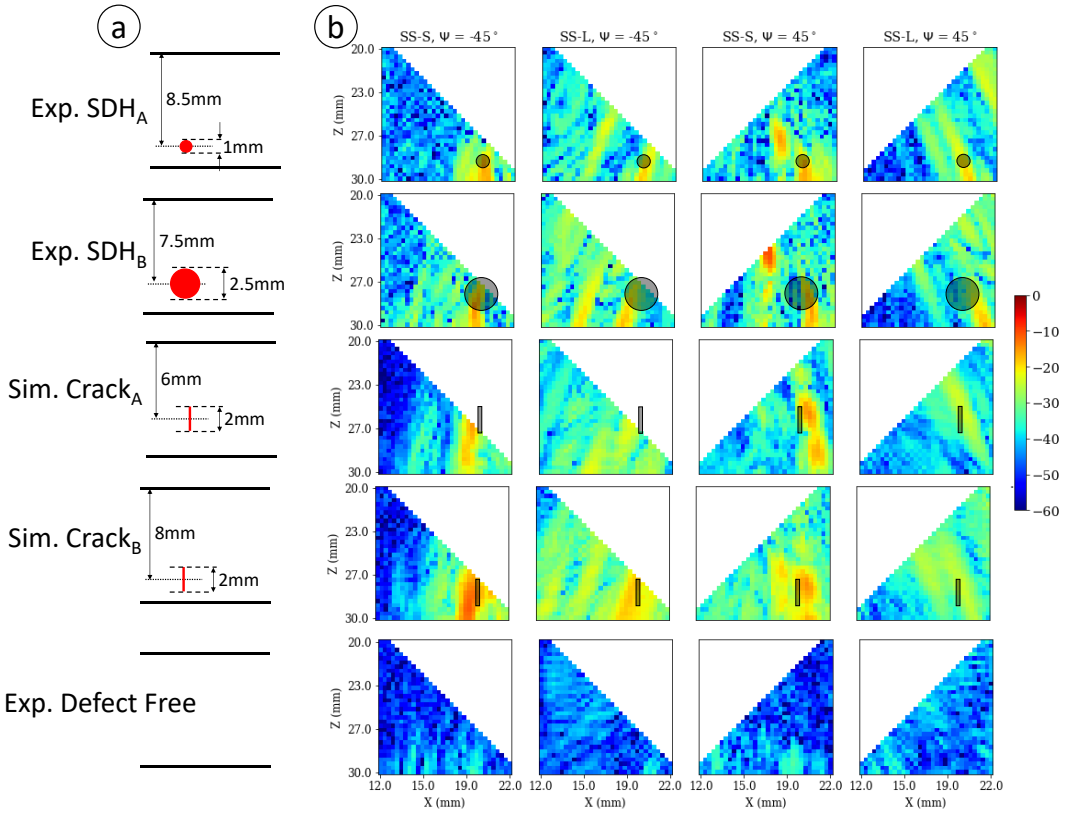

Fig. 2. a) Diagrams and b) sets of example PWI images of defects outside of the training set. The black circles and rectangles in b) show the true size and placement of the defects. All images are on the same dB color scale, normalized to the maximum intensity in the experimental set.

paths other than the one being imaged. Poor angular coverage of a defect from incident and received ray paths blurs indications in images but as PIGs for inline pipe inspection travel at ~2 m/s, capturing data every 1-10 mm, there is too little time to remedy this by firing more than ~3 plane waves per array, per location. However, aleatoric uncertainty is deemed to be negligible in comparison to epistemic uncertainty for this application. This is due to both sources of aleatoric uncertainty being relatively small. Firstly, the data has a large Signal to Noise Ratio (SNR) of ~30 dB. Secondly, while classical sizing methods (such as 6dB drop) suffer due to the weak link between indication size and defect length [9] a CNN can make predictions on more complex features, reducing the need for good angular coverage. If aleatoric uncertainty is not constant across different input samples (i.e. heteroscedastic) it can be estimated by using negative log likelihood as the loss function [43] but this was found to predict values of ~3% of the total uncertainty, supporting the hypothesis of low aleatoric uncertainty. For simplicity, Mean Squared Error (MSE) is used as the loss function in this paper, omitting aleatoric uncertainty from the UQ.

Epistemic uncertainty is the main cause of errors in this application. This is evidenced by the gap in simulated (RMSE = 0.095 mm) and experimental (RMSE = 0.63 mm) test set sizing accuracy of a CNN trained on simulated data. This performance discrepancy is caused by inaccuracies in the simulation such as those given in Table III. Epistemic uncertainty could be reduced by adding experimental data to the training set or using a more accurate simulation. However, these approaches are financially or computationally expensive respectively.

C. *Network Architecture*

Following the work in [9] the CNN architecture used in this

TABLE III
Example sources of epistemic uncertainty for the application in this paper.

Variations in inspection conditions	**Inaccurate simplifications**	**Modes not modelled**
Array mispositioning	Defects modelled as rectangular, perfect reflectors while test set defects have some roughness and rounded tips	Ray paths with more than three legs
Sound speed variation	Surface roughness not modelled	Surface waves
Inconsistency in array element performance	Array assumed to be in far-field of defect in model, but array is partially in defect near field for $L >= 4$ mm	Non-linear effects

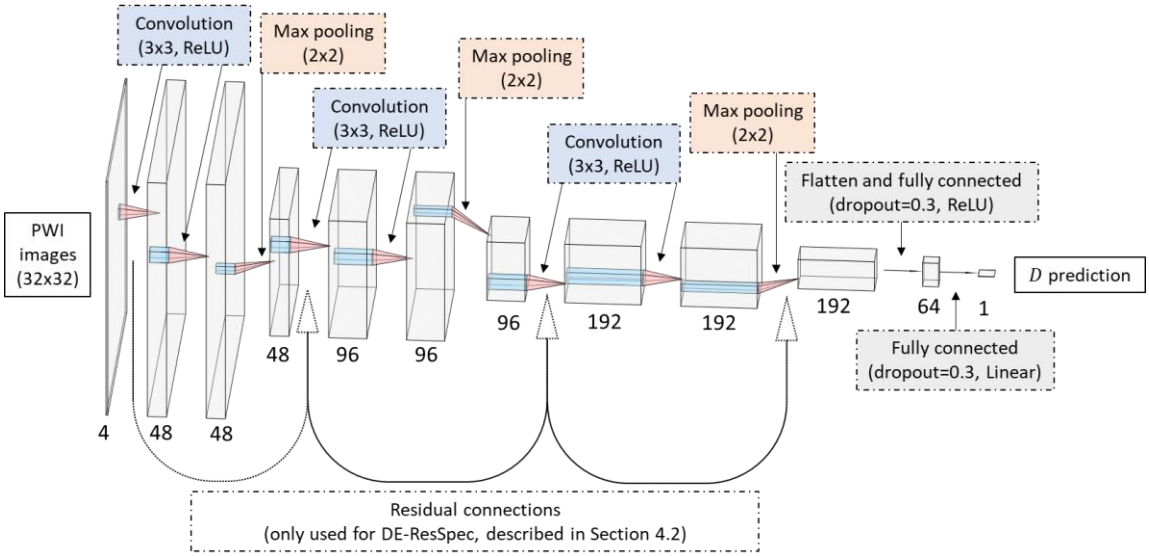

Fig. 3. An illustration of the CNN architecture used throughout the paper.

paper is loosely based on architectures such as VGG-19 due to their widespread success in related image recognition applications. An off-the-shelf architecture is not optimal due to the differences in input size and image content between NDE and visual image applications. As illustrated in Fig. 3, the CNN is composed of repeating blocks of convolutional and down sampling layers, followed by fully connected layers, with Rectified Linear Unit (ReLU) activations used throughout. The convolutional layers aim to perform feature extraction [44], [45], while the fully connected layers predict defect depth, D from those features. Dropout is used after the fully connected layers to reduce overfitting to the training set. The hyperparameters for this network have been iteratively tuned using the validation sets. More details on this design process are presented in [9]. The CNN is trained using the state-of-the-art Adam optimizer [46] with a learning rate of 1×10^{-3}, batch size of 64, and a stopping condition of 50 epochs without a reduction in experimental validation set loss. There are two minor architecture changes from [9] to this paper. Firstly, only a single network is needed to predict D. This matches the structure of the L network in [9]. Secondly, dropout is increased to 0.3, which resulted in slightly better experimental validation set accuracy at the cost of needing ~50 more epochs to converge.

IV. Uncertainty Quantification Methods

To achieve UQ the posterior distribution over the network's weights and biases (W) must be calculated or approximated. Using Bayes' theorem this can be written as,

$$P(W|D) = \frac{likelihood \times prior}{evidence} = \frac{P(D|W)P(W)}{P(D)} = \frac{P(D|W)P(W)}{\int P(D|W)P(W)dW} \quad (1)$$

where D is the training data inputs and outputs. With this, inference for a given input $\hat{x}$ can be calculated by,

$$P(\hat{y}|\hat{x}, D) = \int P(\hat{y}|\hat{x}, W)P(W|D)dW \quad (2)$$

where $\hat{y}$ is the predicted output. However, the posterior is computationally intractable due to the difficulty of evaluating the normalization constant, $P(D) = \int P(D|W)P(W)dW$ due to the high dimensionality of both D and W and the fact that the likelihood, $P(D_i|W)$ and the prior, $P(W)$ are 'nonconjugate' i.e., do not take the same form in relation to W [47]. Approximating this distribution as closely as possible to produce accurate inference of the posterior is the aim of the methods presented in this section.

For all methods considered in this paper the likelihood of the output is considered to be Gaussian,

$$P(\hat{y}|\hat{x}, W) = \mathcal{N}(\mu, \sigma) \quad (3)$$

where both mean, μ and standard deviation, σ are a function of the network's parameters. Because of this assumption, the UQ methods described in this paper can be said to be 'well calibrated' if they demonstrate a 1:1 relationship between predicted uncertainty and σ. Other approaches such as Mixture Density Networks (MDNs) can be used to avoid this assumption, but it is commonly used in deep learning UQ literature, and is considered sufficient for this application.

A. *Deep Ensemble* [20]

Ensembling of machine learning networks has long been recognized as a way to improve accuracy [48], [49], but more recently it has also become a popular UQ method, commonly termed 'Deep Ensembles' (DE) [20]. DE functions by training M networks, usually of the same architecture (as is the case in this paper), to produce a diverse ensemble of predictors. Diversity in the ensemble can be encouraged by training each member with a subset of the full training set, sampled with replacement, this is commonly called bagging or bootstrapping. However, it has been observed that the randomness in network initialisation is sufficient [20], [50] so bagging is not used in

this paper.

The ensemble's overall prediction is represented by a mean (μ) and standard deviation (σ) of the individual member's predictions,

$$\mu = \frac{1}{M}\sum_{i=1}^{M} y_i \tag{4}$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^{M}(y_i - \mu)^2}{M}} \tag{5}$$

where y_i is the output of the i_{th} member of the ensemble, σ is taken as the measure of uncertainty in all methods presented in this paper. The intuition for DE as a UQ method is that different members of the ensemble will tend to output similar values when the inputs are similar to the training data, because each network, even if different, is optimized for that data. But when inputs are less alike to the training data, the networks are more affected by the specificities of the sub-optimal solution reached, producing higher variance results. This can be thought of in a 'loss landscape' perspective as members of the ensemble, due to their different initializations, ending up at local minima, that all accurately predict on the training data, but behave diversely on anomalous data [51]. Prediction error for a specific defect is calculated using

$$Error_j = \mu_j - D_j \tag{6}$$

where j is the index of the defect and D_j is true depth. Error over a full test can be summarised by Root Mean Squared Error (RMSE),

$$RMSE = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(\mu_j - D_j)^2} \tag{7}$$

where N is the size of the test set.

B. Deep Ensemble with Residual Connections [52] *and Spectral Normalization* [53]

Neural networks can suffer from an effect called 'feature-collapse' where distances in the input space are not correlated with distances in the feature space [32]. This means that inputs far from the training data may be mapped close to training set features, erroneously assigning them low uncertainty. It has been shown that feature collapse can be avoided by enforcing 'smoothness' and 'sensitivity' [54]. Smoothness means that small input changes cannot cause large output changes, and sensitivity requires input changes to always change the feature space representation. These properties can be described mathematically by bi-Lipshitz continuity,

$$K_1\|x_1 - x_2\|_2 \leq \|f(x_1) - f(x_2)\|_2 \leq K_2\|x_1 - x_2\|_2 \tag{8}$$

where K_1 and K_2 are the Lipschitz constants of function f and $\|\,.\,\|_2$ represents the L2 norm. In this paper, the feature extractor (convolutional layers) is encouraged to be bi-Lipshitz continuous by spectral normalization [53] and residual connections [52] which create smoothness and sensitivity respectively. Residual networks with spectral normalization have been shown to be 'distance-aware' (i.e. the ability to assess test data's distance from training data distribution) [55] and capture uncertainty effectively [32], [56]. This is explored in this paper as a way to improve the UQ capability of deep ensembles for NDE.

Residual connections create a connection between the input, and layers deeper into a neural network. They were originally proposed to ease the optimization of very deep networks [53] but in doing so they also make the network's activations more sensitive to the input, motivating their use in UQ. As shown in Fig. 3, residual connections take information and shortcut the next few layers by summation with their output. This shortcut should be as close to an identity mapping as possible. As the number of filters changes and max pooling reduces image size by 2 in both width and height, a 1x1 convolutional layer with a stride of 2 and no activation function is used for the residual connections in this paper.

Spectral normalization is equivalent to regularizing the largest singular value of a layer's weight matrix. It has been popularized recently as a way to improve generalization of Generative Adverserial Networks (GANs) [53]. Following [55] and the implementation in [57] the spectral norm, Π, is estimated at every training iteration, for every layer, using the power iteration method. Weights are normalized by multiplication with a scaling constant divided by the spectral norm, $\frac{c_{spectral}}{\Pi}$. This approach has two hyperparameters, the number of power iterations and the scaling constant ($c_{spectral} > 0$). As in [55], one power iteration was found sufficient so is used here and $c_{spectral}$ was set by a grid search for the smallest value that does not reduce the validation set accuracy of network, this was found to be $c_{spectral} = 1.2$. In this paper this method will be referred to as DE-ResSpec from this point onwards.

C. Monte Carlo Dropout [21]

Dropout was originally proposed as a technique for reducing overfitting by setting the output of individual neurons to 0 during training, with probability p, at each iteration [22]. It has later been shown that implementing dropout at both training and test time, before every weight layer, is a close approximation of a deep Gaussian Process [21] and has been termed 'Monte Carlo (MC) dropout'. The intuition for MC dropout as an UQ method is that each initialisation of dropout at test time is acting as a member of an ensemble. As such, μ and σ are calculated using (4) and (5) with M equal to the number of dropout initialisations run at test time, $M_{dropout}$. This is set to 200 in this work as μ and σ were found to change negligibly for $M_{dropout}$ larger than this. Dropout probability, p, is set to 0.3 as larger values significantly increased time to convergence, without improving UQ.

Due to its simplicity, MC dropout has been used in a lot of UQ literature [23] but has also received criticism by [51] in which it is shown to produce significantly less diverse predictors in comparison to DE. This is exampled in [58] where a simple single-hidden layer ReLU network with MC dropout

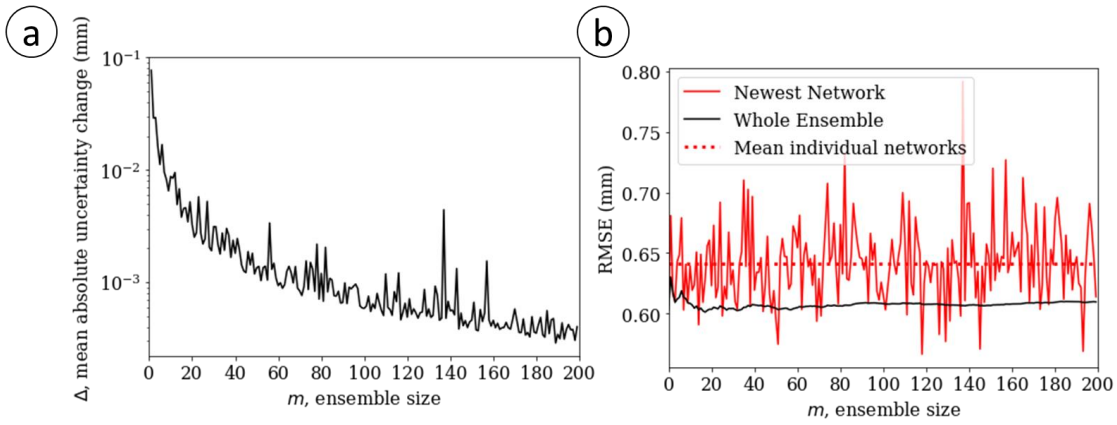

Fig. 4. a) Mean absolute change in uncertainty of the experimental validation set and b) RMSE of the experimental test set for both the whole ensemble and the newest member for increasing ensemble size.

fails to produce high uncertainty between clusters of 2D data. However, the same work also shows that deeper (>= 2 hidden layers) neural networks with MC dropout should theoretically approximate the posterior accurately.

V. Results

This section presents results relating to the quality of UQ from the methods presented in the previous section.

A. Number of networks in ensemble

When originally proposed in [20] it is suggested that five networks are sufficient for effective UQ using DE. However, because neither training nor test time computational resources are limited in this application a larger ensemble can be used. To determine the optimal size of the ensemble, the effect of iteratively adding a network to the ensemble was measured in terms of the mean absolute change in uncertainty,

$$\Delta_m = \frac{1}{N}\sum_{i=1}^{N}\left|u_{m,i} - u_{m+1,i}\right| \tag{9}$$

where $u_{m,i}$ is the uncertainty for the i_{th} sample of the experimental validation set predicted by an ensemble of m networks and N the size of the data set (216 for experimental validation). As shown in Fig. 4a, Δ_m decreases as m increases, indicating a diminishing effect of increasing ensemble size on UQ. 60 networks are used for DE in this paper as $\Delta_{60} \approx 1 \times 10^{-3}\ mm$. This is deemed to be low enough to assume that the ensemble predictions have mostly converged and adding more networks will only minorly change the results.

It should also be noted that while prediction accuracy is not the focus of this paper, ensembling does provide a slight reduction in defect sizing error. This can be seen in Fig. 4b where the experimental test set RMSE of an ensemble with $m >$

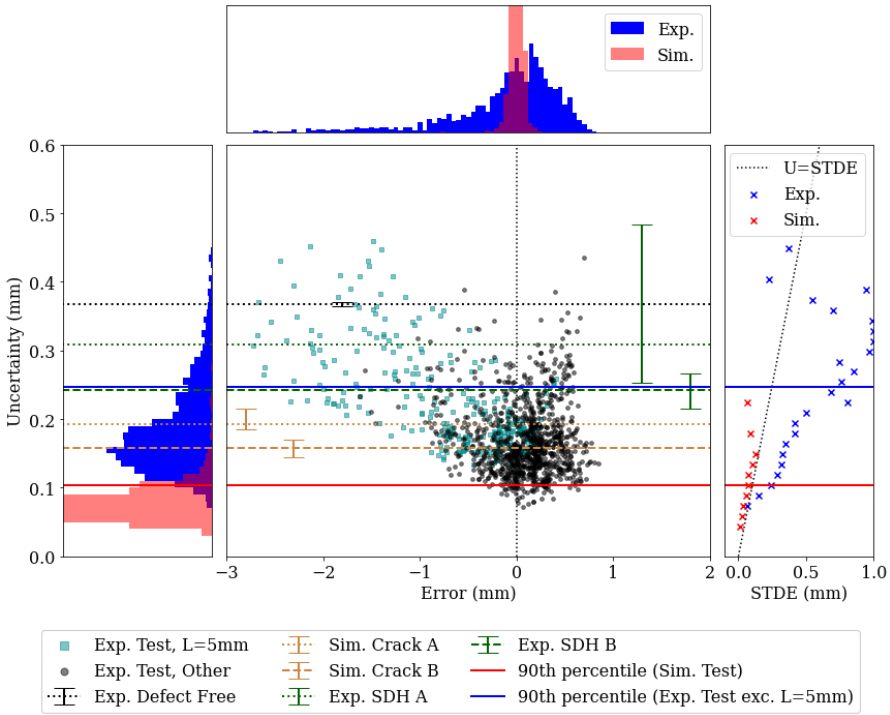

Fig. 5. Deep ensemble (DE) uncertainty predictions for both in and out of distribution test sets. Experimental test set RMSE = 0.592 mm.

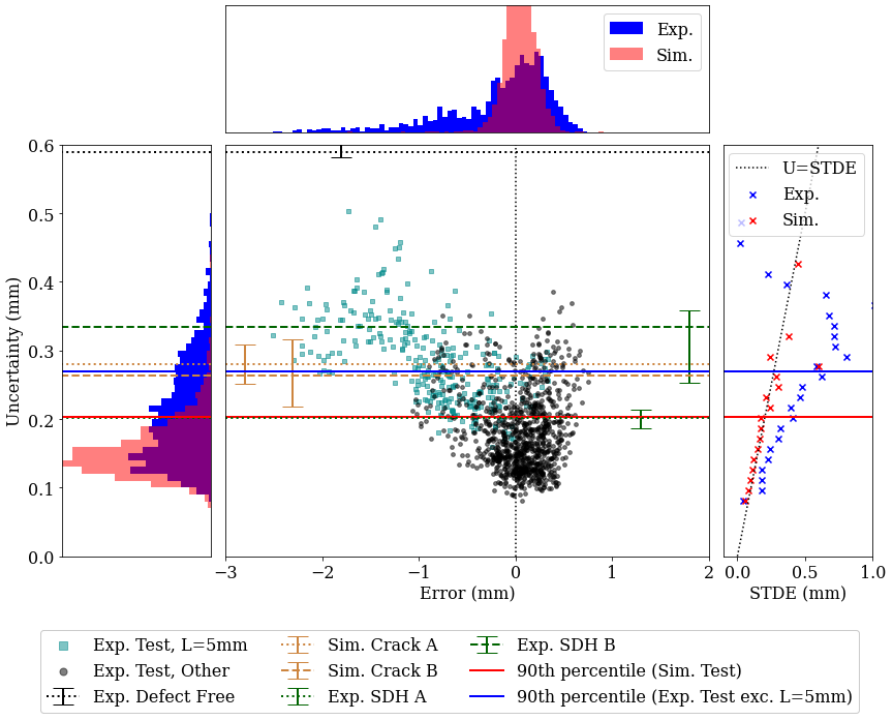

Fig. 6. Deep ensemble with residual connections and spectral normalization (DE ResSpec) uncertainty predictions for both in and out of distribution test sets. Experimental test set RMSE = 0.5831 mm.

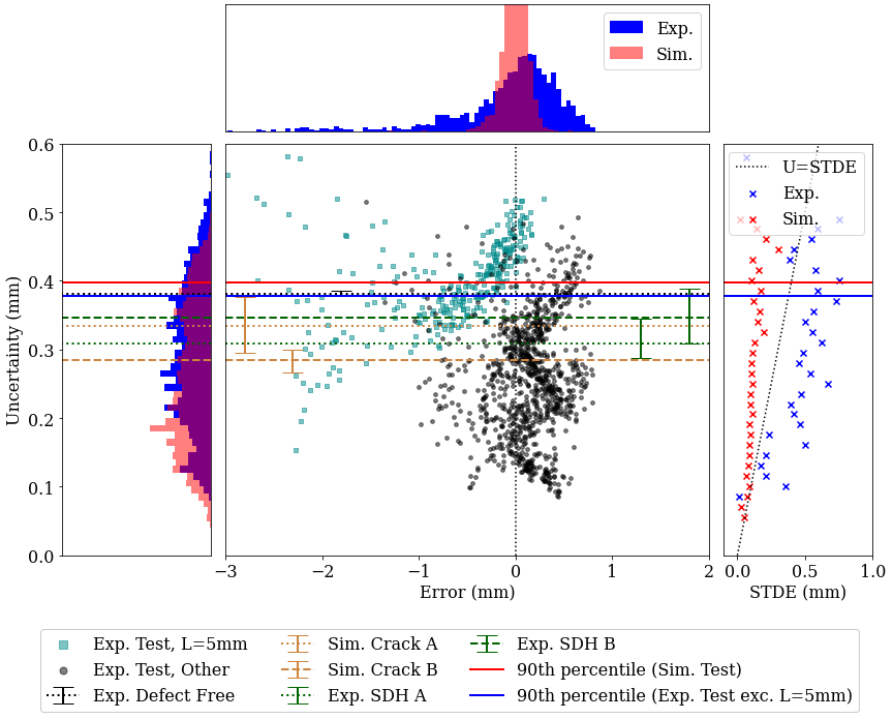

Fig. 7. Monte Carlo (MC) dropout uncertainty predictions for both in and out of distribution test sets. Experimental test set RMSE over 30 initialisations = 0.673±0.05 mm.

10 (solid black line) is ~0.035 mm lower than the mean RMSE of the 200 networks when used independently (dotted red line).

B. Calibration

The uncertainty quantification (Eq. 5) and prediction error (Eq. 6) of the methods described in Section IV are illustrated in Figs. 5-7. The predictions for uncertainty and crack depth (D) for DE and DE ResSpec (Figs. 5,6) are formed from 60 independently trained networks. For MC dropout (Fig. 7) inference uses the output of one network with 200 forward passes, assigning a new random seed to the dropout realizations each time. The main scatter plots in these figures show predicted uncertainty vs. sizing error for each defect in the experimental test set. Effective UQ for in distribution data in this plot appears as a zero-mean distribution of error that widens

TABLE IV
Metrics regarding linear fit of STDE to uncertainty below 90th percentile of uncertainty predictions for simulated and experimental test sets

	Simulated Test Set		*Experimental Test Set*	
	R	*Mean(U-STDE) (mm)*	*R*	*Mean(U-STDE) (mm)*
DE	0.98	0.032	0.95	-0.24
DE-ResSpec	0.99	0.015	0.98	-0.15
MC Dropout	0.84	0.11	0.84	-0.21

as uncertainty increases. Also shown in main scatter plot, as horizontal dotted and dashed lines, are the calculated uncertainties for OOD datasets. These do not have associated error as there is no equivalent 'true' value. Error bars for this data show the 25th and 75th percentiles of uncertainty across the full range of X-positions for each defect type. Ideally, these data should be assigned higher uncertainty values than most in-distribution data. Blue bars in the histograms plotted above and to left show the uncertainty and sizing error distributions for experimental and simulated test datasets based on bins of width of 0.05 mm (above) and 0.01 mm (left). For visual clarity, the red simulated test data histograms are not shown in the main scatter plot. Solid horizontal lines indicate the 90th percentile of the test sets' UQ. Graphs plotted to the right show aggregated uncertainty vs standard deviation of error (STDE). These are calculated by splitting the uncertainty predictions into equally spaced bins of height 0.015 mm and calculating the STDE in each bin containing more than one defect. The black dotted line is uncertainty = STDE which is the ideal result for in-distribution data as points on this line indicate predicted uncertainty is close to σ (as defined in (3)). Table IV gives correlation coefficient, R for the linear fits to the data in the right-most graphs as well as the mean difference between STDE and predicted uncertainty. While, for the methods described in this paper, the relationship between STDE and uncertainty is expected to be monotonic, there is no guarantee it will be 1:1, or even linear. Therefore, the following sections describe the observed trends in calibration of uncertainty to error for the experimental and simulated test sets.

1) Simulated

Below the 90th percentile of sim test, DE (Fig. 5) and DE-ResSpec (Fig. 6) have a strong linear relationship between uncertainty and STDE. This is quantified by the high correlation coefficient of linear fits, ($R_{DE,Sim} = 0.99$, $R_{DE-RS,Sim} = 0.98$). The lines fit to this data have a slope of ~1 for both methods with low mean differences between uncertainty and STDE of 0.032 mm for DE and 0.015 mm for DE-ResSpec. In the upper tail of the uncertainty distribution (upper 10th percentile of sim test) both methods show increased scatter in STDE. This is likely due to the low amount of data in the STDE bins. MC dropout (Fig. 7) produces a linear fit for the simulated test set ($R_{MC,Sim}$ = 0.84) but its slope is 2.3, severely underestimating error for larger uncertainty values.

2) Experimental

In the upper tail of the uncertainty distribution (upper 10th percentile of exp. test) both DE and DE-ResSpec underestimate error significantly. While this is likely contributed to by insufficient ensemble diversity it is mainly due to inaccuracies in the simulation of the $L = 5$ mm defects. This is because the simulation used to create the training set assumes that the receiving transducer array elements are in the far-field of the defect. This is not the case for the $L = 5$ mm defects, noticeably effecting their PWI images [9]. As shown in Figs. 5-7, the experimental defects of length $L = 5$ mm are significantly undersized because of this domain shift. However, with DE and DE-ResSpec they are also assigned higher uncertainty. DE-ResSpec achieves this most effectively, assigning a mean uncertainty to $L = 5$ mm defects higher than 92% of the rest of the experimental test set. Even without knowing the true size of the defects this would highlight to the operator that they are somehow seen as anomalous by the networks. However, these uncertainty values are still low in comparison to their absolute error. This is because the difference in simulated and experimental $L = 5$ mm defects creates a systematic undersizing in all members of the ensemble. As this change in the predictions has a non-zero mean across the ensemble the increased uncertainty is not full captured in the ensemble's overall variance (Eq. 5). This is an example of domain shift negatively affecting the quality of UQ, a known issue [59].

Experimental test set uncertainty below the 90th percentile increases monotonically with STDE for both DE and DE-ResSpec ($R_{DE,Exp} = 0.95$, $R_{DE-RS,Exp} = 0.98$) whereas MC dropout shows more significant scatter ($R_{MC,Exp} = 0.84$). All three of these trends have a slope <1, indicating that UQ is significantly underestimating error. The consequence of this for implementation of these methods is that if uncertainty predictions are to be used as an estimate of expected sizing error on a new experimental sample, an experimental validation set is needed to calculate the slope. This method is commonly called 'temperature scaling' [60]. However, even without temperature scaling, the strong linear fit means that higher uncertainty is a strong indicator of higher error for the DE based approaches.

3) Anomaly Detection

Effective UQ should detect test cases drawn from distributions significantly far away from that of the training set. As the network has little to no prior information about these cases, it should assign them high uncertainty. As described in Section III.B.2 this is primarily tested here using defect types not included in the training set. All three methods assign higher

uncertainty to the OOD defects than the bulk of the experimental test set but for MC dropout it is also almost all below the 90th percentile of simulated data, demonstrating poor anomaly detection. DE-ResSpec demonstrates the best anomaly detection; assigning uncertainty above 90% of the non $L = 5$ mm experimental test set to 60% of the OOD cases. Exp. SDH A is assigned the lowest uncertainty by DE-ResSpec. This makes intuitive sense, as of all the OOD defects, it is the smallest and nearest the back wall, and therefore produces PWI images that most closely resemble a surface breaking crack. This is exampled in Fig. 2b in comparison to Fig. 1d.

4) Choosing an Uncertainty Threshold

In implementing these UQ methods for industry, test cases with uncertainty above a certain threshold can be dealt with separately. This may mean inspection by a human operator, further data acquisition, use of traditional sizing methods or a combination of these approaches. To do this, a value for the uncertainty threshold must be decided upon. Ideally, this would be done through the use of an experimental validation set that represents the true inspection conditions well. However, in the absence of such data, using the simulated validation set could be an effective approach. The left and top panels of Fig. 6 show that this works well for DE-ResSpec as both the simulated and experimental 'in distribution' test cases are assigned similar uncertainty distributions, meaning that almost all high sizing error (>1 mm) and OOD cases are above the 90th percentile of the simulated validation set. In contrast, in Fig. 5, DE demonstrates limited overlap between the UQ distributions for simulated and experimental test sets. This means that using a cutoff defined by only simulated data will find almost all experimental data anomalous with DE. It is hypothesized that the regularization of the spectral norm is responsible for DE-ResSpec demonstrating better simulated and experimental overlap than DE. MC-Dropout has good overlap but doesn't distinguish either of these sets from OOD data.

VI. Making Efficient Use of Resources

In the application considered in this paper the computational resources at both training and test time are not a barrier for implementation of DE. Both the training and testing of are trivially parallelizable but even with multiple GPUs, some applications require more computational efficiency. This section discusses ways that training and inference time for DE can be reduced.

A. Training Resources

As the architecture used here has a relatively low number of parameters (842,000) each epoch takes ~3.5s using a NVIDIA GeForce GTX 1070 Ti, so training a full ensemble of 60 networks can be completed in ~6hrs. If a more complex network was used (e.g. VGG 19 with 138 million) training an ensemble could take multiple weeks, making the development cycle very slow. Alongside its simplicity, MC dropout has also gained popularity as an UQ method because it only requires the training of one network so is a good candidate for reducing training time. Another approach is 'snapshot ensembles' [61] in which the members of an ensemble can be captured from one initialisation, using a cyclic learning rate. For this application snapshot ensembles were found to provide significantly worse UQ than DE. It is hypothesized that this is because the local optima found by snapshot ensembles are not as diverse as that found by re-initializing the network's parameters.

B. Test Resources

Inference with the 60-network ensemble used in this paper takes $\sim 8ms$ per image set which for most applications is quick enough to be considered 'realtime.' However, if realtime inference was required on lightweight hardware and/or using a more complex network the test time resources would need to be managed more efficiently. This could be achieved by pruning the weights of the individual networks [62], using a smaller number of networks in the ensemble by optimizing which members are used [63] or distilling the ensemble down to a single 'multi-headed' network with one set of common convolutional layers and multiple sets of fully connected layers [64].

VII. Conclusions

This paper has investigated the performance of UQ using DE, DE-ResSpec and MC Dropout for modern deep learning in application to inline pipe inspection when using a simulated training set and experimental test data. The success of these methods is judged by their calibration and anomaly detection performance. MC Dropout demonstrates only slightly raised uncertainty values for OOD samples and poorly calibrated uncertainty estimates. DE-ResSpec produced the best calibration on simulated test data, created the largest gap between in-distribution and out-of-distribution data and is the most reliable method in terms of assigning high uncertainty to high error test cases. However, while both DE and DE-ResSpec show a strong linear fit between experimental data error and uncertainty, the gradient of this fit is << 1, meaning that uncertainty significantly underestimates error. The implication of this for industrial applications is that an experimental validation set for scaling is needed if uncertainty values are used to infer expected prediction error. However, as the monotonic relationship between uncertainty and error is strong, even without an experimental validation set, predicted uncertainty can be used to compare relative error between test cases and detect anomalies. It is therefore the opinion of the authors that DE-ResSpec is currently the most appropriate method for UQ when using deep learning for NDE.

One of the biggest unknowns in the field of data science for NDE is how data-driven NDE inspections are to be qualified. Within the current industrial framework, physics-based data analysis is qualified on a small pool of test samples and generalization assured by the interpretability of the method. However, in the future, the high levels of accuracy demonstrated by 'black-box' methods may well create a drive to qualify them by rigorous testing on a large range of test

samples. For this to be realized, UQ methods such as the ones presented in this paper, are going to be essential. As presented in this paper, DE and DE-ResSpec are suitable for application to approximating uncertainty of deep learning for NDE. Improvements could be made by research into producing better calibrated UQ on experimental test data, despite the domain shift from the simulated training set. Domain adaptation methods or techniques for increasing the diversity within the ensemble are promising candidates for this problem.

Appendix

Supporting code and data are available at the University of Bristol data repository, data.bris, at https://doi.org/10.5523/bris.xpeoi1k840fj2mrvfuaejo2t1.

References

[1] L. Udpa and S. S. Udpa, "Neural networks for the classification of nondestructive evaluation signals," *IEE Proceedings, Part F: Radar and Signal Processing*, vol. 138, no. 1, pp. 41–45, 1991, doi: 10.1049/ip-f-2.1991.0007.

[2] N. Amiri, G. H. Farrahi, K. R. Kashyzadeh, and M. Chizari, "Applications of ultrasonic testing and machine learning methods to predict the static & fatigue behavior of spot-welded joints," *Journal of Manufacturing Processes*, vol. 52, pp. 26–34, Apr. 2020, doi: 10.1016/j.jmapro.2020.01.047.

[3] M. Mishra, A. S. Bhatia, and D. Maity, "Predicting the compressive strength of unreinforced brick masonry using machine learning techniques validated on a case study of a museum through nondestructive testing," *Journal of Civil Structural Health Monitoring*, pp. 1–15, Mar. 2020, doi: 10.1007/s13349-020-00391-7.

[4] Z. Lin, H. Pan, G. Gui, and C. Yan, "Data-driven structural diagnosis and conditional assessment: from shallow to deep learning," in *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2018*, Mar. 2018, vol. 10598, p. 38. doi: 10.1117/12.2296964.

[5] J. Ye, S. Ito, and N. Toyama, "Computerized ultrasonic imaging inspection: From shallow to deep learning," *Sensors (Switzerland)*, vol. 18, no. 11, Nov. 2018, doi: 10.3390/s18113820.

[6] I. Virkkunen, T. Koskinen, O. Jessen-Juhler, and J. Rinta-Aho, "Augmented ultrasonic data for machine learning," *Journal of Nondestructive Evaluation*, vol. 40, no. 1, pp. 1–11, 2021.

[7] S. Sambath, P. Nagaraj, and N. Selvakumar, "Automatic defect classification in ultrasonic NDT using artificial intelligence," *J Nondestr Eval*, vol. 30, no. 1, pp. 20–28, 2011.

[8] X. L. Travassos, S. L. Avila, and N. Ida, "Artificial neural networks and machine learning techniques applied to ground penetrating radar: A review," *Applied Computing and Informatics*, 2020.

[9] R. J. Pyle, R. L. T. Bevan, R. R. Hughes, R. K. Rachev, A. Ait Si Ali, and P. D. Wilcox, "Deep Learning for Ultrasonic Crack Characterization in NDE," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 68, no. 5, pp. 1854–1865, 2020, doi: 10.1109/TUFFC.2020.3045847.

[10] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: A brief review," *Comput Intell Neurosci*, vol. 2018, 2018.

[11] A. S. Lundervold and A. Lundervold, "An overview of deep learning in medical imaging focusing on MRI," *Zeitschrift für Medizinische Physik*, vol. 29, no. 2, pp. 102–127, 2019.

[12] Dipl.-P. Marija Bertović, "Human Factors in Non-Destructive Testing (NDT): Risks and Challenges of Mechanised NDT," 2016.

[13] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[14] A. Bernieri, L. Ferrigno, M. Laracca, and M. Molinara, "Crack shape reconstruction in Eddy current testing using machine learning systems for regression," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 9, pp. 1958–1968, 2008, doi: 10.1109/TIM.2008.919011.

[15] G. Psuj, "Multi-sensor data integration using deep learning for characterization of defects in steel elements," *Sensors (Switzerland)*, vol. 18, no. 1, Jan. 2018, doi: 10.3390/s18010292.

[16] D. Medak, L. Posilović, M. Subašić, M. Budimir, and S. Lončarić, "Automated Defect Detection from Ultrasonic Images Using Deep Learning," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 2021.

[17] J. Sresakoolchai and S. Kaewunruen, "Detection and Severity Evaluation of Combined Rail Defects Using Deep Learning," *Vibration*, vol. 4, no. 2, pp. 341–356, 2021.

[18] M.-H. DOD, "Department of Defense Handbook: Nondestructive Evaluation System Reliability Assessment," *Department of Defense, Washington, DC*, 2009.

[19] A. P. Dawid, "The well-calibrated Bayesian," *J Am Stat Assoc*, vol. 77, no. 379, pp. 605–610, 1982.

[20] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," *arXiv preprint arXiv:1612.01474*, 2016.

[21] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, 2016, pp. 1050–1059.

[22] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[23] M. Abdar *et al.*, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," *Information Fusion*, 2021.

[24] P. Zhu, Y. Cheng, P. Banerjee, A. Tamburrino, and Y. Deng, “A novel machine learning model for eddy current testing with uncertainty,” *NDT & E International*, vol. 101, pp. 104–112, 2019.

[25] S. O. Sajedi and X. Liang, “Uncertainty-assisted deep vision structural health monitoring,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 36, no. 2, pp. 126–142, 2021.

[26] C. M. Bishop, “Mixture density networks,” 1994.

[27] I. D. Khurjekar and J. B. Harley, “Uncertainty aware deep neural network for multistatic localization with application to ultrasonic structural health monitoring,” *arXiv preprint arXiv:2007.06814*, 2020.

[28] W. Chen, Y. Gao, L. Gao, and X. Li, “A new ensemble approach based on deep convolutional neural networks for steel surface defect classification,” *Procedia CIRP*, vol. 72, pp. 1069–1072, 2018.

[29] M. Marino, K. Virupakshappa, and E. Oruklu, “A Stacked Ensemble Neural Network Classifier for Ultrasonic Non-Destructive Evaluation Applications,” in *2020 IEEE International Ultrasonics Symposium (IUS)*, 2020, pp. 1–4.

[30] F. Chang, M. Liu, M. Dong, and Y. Duan, “A mobile vision inspection system for tiny defect detection on smooth car-body surfaces based on deep ensemble learning,” *Measurement Science and Technology*, vol. 30, no. 12, p. 125905, 2019.

[31] J. Bradshaw, A. G. de G. Matthews, and Z. Ghahramani, “Adversarial examples, uncertainty, and transfer testing robustness in gaussian process hybrid deep networks,” *arXiv preprint arXiv:1707.02476*, 2017.

[32] J. van Amersfoort, L. Smith, A. Jesson, O. Key, and Y. Gal, “On Feature Collapse and Deep Kernel Learning for Single Forward Pass Uncertainty,” *arXiv preprint arXiv:2102.11409*, 2021.

[33] G. E. Hinton and D. Van Camp, “Keeping the neural networks simple by minimizing the description length of the weights,” in *Proceedings of the sixth annual conference on Computational learning theory*, 1993, pp. 5–13.

[34] A. Graves, “Practical variational inference for neural networks,” *Adv Neural Inf Process Syst*, vol. 24, 2011.

[35] J. Hensman, A. Matthews, and Z. Ghahramani, “Scalable variational Gaussian process classification,” in *Artificial Intelligence and Statistics*, 2015, pp. 351–360.

[36] S. Kullback and R. A. Leibler, “On information and sufficiency,” *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[37] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *arXiv preprint arXiv:1312.6114*, 2013.

[38] L. Le Jeune, S. Robert, E. L. Villaverde, and C. Prada, “Plane Wave Imaging for ultrasonic non-destructive testing: Generalization to multimodal imaging,” *Ultrasonics*, vol. 64, pp. 128–138, 2016.

[39] P. D. Wilcox and A. Velichko, “Efficient frequency-domain finite element modeling of two-dimensional elastodynamic scattering,” *J Acoust Soc Am*, vol. 127, no. 1, pp. 155–165, Jan. 2010, doi: 10.1121/1.3270390.

[40] R. K. Rachev, P. D. Wilcox, A. Velichko, and K. L. McAughey, “Plane Wave Imaging Techniques for Immersion Testing of Components with Non-Planar Surfaces,” *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, pp. 1–1, Jan. 2020, doi: 10.1109/tuffc.2020.2969083.

[41] L. W. Schmerr, *Fundamentals of ultrasonic nondestructive evaluation*. Springer, 2016.

[42] H. A. Bloxham, A. Velichko, and P. D. Wilcox, “Combining simulated and experimental data to simulate ultrasonic array data from defects in materials with high structural noise,” *IEEE Trans Ultrason Ferroelectr Freq Control*, vol. 63, no. 12, pp. 2198–2206, 2016.

[43] A. Kendall and Y. Gal, “What uncertainties do we need in bayesian deep learning for computer vision?,” in *Advances in neural information processing systems*, 2017, pp. 5574–5584.

[44] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, “Return of the devil in the details: Delving deep into convolutional nets,” *arXiv preprint arXiv:1405.3531*, 2014.

[45] J. Donahue *et al.*, “Decaf: A deep convolutional activation feature for generic visual recognition,” in *International conference on machine learning*, 2014, pp. 647–655.

[46] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” Dec. 2015.

[47] C. M. Bishop, “Pattern recognition,” *Mach Learn*, vol. 128, no. 9, 2006.

[48] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[49] T. G. Dietterich, “Ensemble methods in machine learning,” in *International workshop on multiple classifier systems*, 2000, pp. 1–15.

[50] S. Lee, S. Purushwalkam, M. Cogswell, D. Crandall, and D. Batra, “Why M heads are better than one: Training a diverse ensemble of deep networks,” *arXiv preprint arXiv:1511.06314*, 2015.

[51] S. Fort, H. Hu, and B. Lakshminarayanan, “Deep ensembles: A loss landscape perspective,” *arXiv preprint arXiv:1912.02757*, 2019.

[52] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[53] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, “Spectral normalization for generative adversarial networks,” *arXiv preprint arXiv:1802.05957*, 2018.

[54] J. Van Amersfoort, L. Smith, Y. W. Teh, and Y. Gal, “Uncertainty estimation using a single deep deterministic neural network,” in *International Conference on Machine Learning*, 2020, pp. 9690–9700.

[55] J. Z. Liu, Z. Lin, S. Padhy, D. Tran, T. Bedrax-Weiss, and B. Lakshminarayanan, “Simple and principled uncertainty estimation with deterministic deep

learning via distance awareness," *arXiv preprint arXiv:2006.10108*, 2020.

[56] J. Mukhoti, A. Kirsch, J. van Amersfoort, P. H. S. Torr, and Y. Gal, "Deterministic neural networks with appropriate inductive biases capture epistemic and aleatoric uncertainty," *arXiv preprint arXiv:2102.11582*, 2021.

[57] Tensorflow, "tfa.layers.SpectralNormalization," 2021. https://www.tensorflow.org/addons/api_docs/python/tfa/layers/SpectralNormalization (accessed Sep. 08, 2021).

[58] A. Y. K. Foong, D. R. Burt, Y. Li, and R. E. Turner, "On the expressiveness of approximate inference in bayesian neural networks," *arXiv preprint arXiv:1909.00719*, 2019.

[59] Y. Ovadia *et al.*, "Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift," *arXiv preprint arXiv:1906.02530*, 2019.

[60] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International Conference on Machine Learning*, 2017, pp. 1321–1330.

[61] G. Huang, Y. Li, G. Pleiss, Z. Liu, J. E. Hopcroft, and K. Q. Weinberger, "Snapshot ensembles: Train 1, get m for free," *arXiv preprint arXiv:1704.00109*, 2017.

[62] U. Dorjsembe, J. H. Lee, B. Choi, and J. W. Song, "Sparsity Increases Uncertainty Estimation in Deep Ensemble," *Computers*, vol. 10, no. 4, p. 54, 2021.

[63] R. Hu, Q. Huang, S. Chang, H. Wang, and J. He, "The MBPEP: a deep ensemble pruning algorithm providing high quality uncertainty prediction," *Applied Intelligence*, vol. 49, no. 8, pp. 2942–2955, 2019.

[64] L. Tran *et al.*, "Hydra: Preserving ensemble diversity for model distillation," *arXiv preprint arXiv:2001.04694*, 2020.

Richard J. Pyle was born in Torquay, U.K. in 1996. He received an M.Eng. degree in mechanical engineering from The University of Bristol, U.K. in 2018.

Through the summer of 2017 he worked for Cavendish Nuclear as a graduate design engineer. He is now studying for an Eng.D. degree in ultrasonic phased array signal processing at The University of Bristol, sponsored by Baker Hughes, Cramlington, U.K. His current research interests include phased array imaging, data compression, defect characterization and machine learning.

Robert R. Hughes was born in Bristol, U.K., in 1989. He received an M.Phys. degree in physics followed by an Engineering Doctorate (Eng.D.) in non-destructive evaluation from the Department of Physics, University of Warwick, in 2016. His Eng.D. research was sponsored by Rolls-Royce plc., Bristol, where he carried out an industrial placement between 2014 and 2015 focusing on eddy-current array sensor development and data-analysis.

In 2015, Dr. Hughes took up a position as Research Associate with the Department of Mechanical Engineering, University of Bristol, U.K, where he developed eddy-current inspection and data-analysis techniques for characterising surface-breaking defects and carbon-fibre composite structures. From 2019, Dr. Hughes has been a Lecturer in non-destructive testing at the Department of Mechanical Engineering, University of Bristol, U.K where his current research interests include eddy-current inspection, inversion of inhomogenous materials, defect characterisation and advanced data-analysis techniques, as well as magnetic particle sensing & manipulation in microfluidic environments.

Amine Ait Si Ali received the Dipl.-Ing. (M.Eng.) degree in computer science from the University of Science and Technology Houari Boumediene, Algiers, Algeria, in 2009, the M.Sc. degree in embedded intelligent systems from the University of Hertfordshire, Hatfield, U.K, in 2012, and the Ph.D. degree in computer science from the University of the West of Scotland, Paisley, U.K., in 2016.

He took many research positions, including Research Assistant with the School of Engineering, Qatar University, Doha, Qatar and KTP Associate with the Department of Computer and Information Sciences, University of Northumbria, Newcastle upon Tyne, U.K. He is currently a Data Scientist with Process & Pipeline Services, Digital Solutions, Baker Hughes, Cramlington, U.K. His research interests are mainly in machine learning, big data, cloud computing, non-destructive testing, connected health and custom computing using FPGAs and heterogeneous embedded systems

Paul D. Wilcox was born in Nottingham (England) in 1971. He received an M.Eng. degree in Engineering Science from the University of Oxford (Oxford, England) in 1994 and a Ph.D. from Imperial College (London, England) in 1998. He remained in the Non-Destructive Testing (NDT) research group at Imperial College as a Research Associate until 2002, working on the development of guided wave array transducers for large area inspection.

Since 2002 Prof. Wilcox has been with the Department of Mechanical Engineering at the University of Bristol (Bristol, England) where his current title is Professor of Dynamics. He held an EPSRC Advanced Research Fellowship in Quantitative Structural Health Monitoring from 2007 to 2012, was Head of the Mechanical Engineering Department from 2015 to 2018, and has been a Fellow of the Alan Turing Institute for Data Science since 2018. In 2015 he was a co-founder of Inductosense Ltd., a spin-out company which is commercialising inductively-coupled embedded ultrasonic sensors. His research interests include array transducers, embedded sensors, ultrasonic particle manipulation, long-range guided wave inspection, structural health monitoring, elastodynamic scattering and signal processing.