Domain Adapted Deep-Learning for Improved Ultrasonic Crack Characterization Using Limited Experimental Data

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

***Abstract*— Deep-learning is an effective method for ultrasonic crack characterization due to its high level of automation and accuracy. Simulating the training set has been shown to be an effective method of circumventing the lack of experimental data common to non-destructive evaluation applications. However, a simulation can neither be completely accurate, nor capture all variability present in the real inspection. This means that the experimental and simulated data will be from different (but related) distributions, leading to inaccuracy when a deep learning algorithm trained on simulated data is applied to experimental measurements. This paper aims to tackle this problem through the use of Domain Adaptation (DA).**

A convolutional neural network is used to predict the depth of surface-breaking defects, with inline pipe inspection as the targeted application. Three DA methods across varying sizes of experimental training data are compared to two non-DA methods as a baseline. The performance of the methods tested is evaluated by sizing 15 experimental notches of length 1-5 mm and inclined at angles of up to 20° from the vertical. Experimental training sets are formed with between 1 and 15 notches. Of the DA methods investigated, an adversarial approach is found to be the most effective way to use the limited experimental training data. With this method, and only three notches, the resulting network gives a Root Mean Square Error (RMSE) in sizing of 0.5 ± 0.037 mm whereas with only experimental data RMSE is 1.5 ± 0.13 mm and with only simulated data it is 0.64 ± 0.044 mm.

***Index Terms*— Domain adaptation, deep-learning, neural networks, plane wave imaging, simulation, ultrasound, defect characterization**

I. Introduction

NON-Destructive Evaluation (NDE) techniques are used to assess the integrity of a component with the aim of extending its lifespan, reducing manufacturing costs and improving overall safety. The NDE application targeted in this paper is ultrasonic inline-pipe inspection in which transducers mounted on a 'pig' (pipeline inspection gauge) are used to detect defects as the pig travels along the pipeline, in the flow of the product, capturing data every 1-10 mm. When onboard processing detects a defect, the data is compressed and stored for offline analysis with the parameter of interest usually being the remaining thickness of intact pipe. Inferring the health of a component from its response to a stimulus, such as ultrasound or x-ray, is commonly called the inverse problem. For inline pipe-inspection (as in most NDE applications), this is classically solved by a skilled human operator inspecting the data. However, the success of machine learning in related fields such as medical imaging [1] has drawn the attention of NDE researchers to the possibility of using data driven methods to improve defect characterisation, relative to classical methods. Automated data analysis such as this can provide significant cost savings by reducing required operator time and thus the changes of incorrect sentencing caused by human factors [2].

While this emerging field has seen a large number of successes, with machine learning techniques demonstrating human-level NDE data interpretation [3]–[10], there are still essential challenges to overcome. The three most important of these are choosing effective parameters to learn from (feature engineering), the scarcity of data, and mistrust in the 'black box' nature of machine learning [11]. The first of these challenges can be solved by inputting raw data to the machine learning algorithm, therefore tasking it with performing both feature engineering and inference. This approach is commonly called 'deep-learning'. However, whilst solving the first problem, deep-learning exacerbates the second, as vast amounts of data (order 10^4) are required to train deep networks. In some very niche NDE applications enough experimental data is available [12], but this is not often the case as the data of interest is usually from defective components, which are expensive to manufacture. One solution to this problem is to increase the size of the training set via data augmentation methods such as cropping, translating, flipping, etc. [9], [13]–[15]. However, while these augmentations produce realistic examples for photographic images (where these methods are commonly used [16]) this is not necessarily true for NDE modalities. Recent research explores NDE specific data augmentation methods such as shifting signals in the time domain, coupled with random amplitude multipliers [17]. However, as both the

Submitted for review on 11/01/22. 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, Rhodri Bevan, 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

indication's shape and amplitude can change with defect size, effective data augmentation still remains an unsolved problem for large and complex defect types. Another solution to the data shortage problem is to use a physics-based model to simulate the deep-learning training set. This approach has been used recently to train modern deep learning architectures for defect detection using X-ray data [18] as well as for ultrasonic sizing of surface breaking defects using a Convolutional Neural Network (CNN) [19], demonstrating sizing almost four times better than the conventional '6dB drop' method [20].

However, while simulating a training set is an attractive approach, simulated NDE data can never perfectly match real data as it invariably contains simplifications and assumptions. This means that a model trained only with simulated data may not accurately size experimental data. This paper looks to solve this problem by including a small pool of experimental data in the training process. This is a 'Transfer Learning' (TL) [21] problem in that it aims to train a network using data from a 'source' domain (i.e. simulation), that is intended to perform a task in a different, but related, 'target' domain (i.e. experiment). TL for problems with the same task in both domains, as in this paper, is called Domain Adaptation (DA). Note that to avoid confusion 'model' is used exclusively to refer to physics-based forward models while 'network' is used to refer to machine learning predictors. For machine learning terminology and definitions see [22].

In this paper three DA approaches are presented and compared against two baseline cases in their ability to improve the sizing accuracy of a CNN by adding a small amount of experimental data to the simulated training set. Building on the work in [20] the same CNN architecture, inspection set up, simulation methodology and imaging protocol are used here. Also, as with [20], the DA methods presented are applicable to any NDE application and modality but their effectiveness is demonstrated here by considering inline pipe inspection. The pig considered uses a ring of ultrasonic arrays to induce plane waves in the pipe that travel at both 45° and -45° to the surface. From the received data four distinct ultrasonic array images are created for each surface breaking defect and used as input to the CNN to predict the through thickness extent of the defect (from here on referred to as 'crack depth'). The effectiveness of the baseline and DA methods to improve the CNN's sizing accuracy is explored in this paper by training with a simulated training set size of 14,343 and a varying size of experimental training set (54-729, from measurements on 1-14 physical defect samples). The sizing accuracy of the resulting network is assessed using an experimental test set formed of 756 image sets from 15 physical defect samples not included in the training set.

The rest of this paper is structured as follows. Section II outlines previous, relevant research, Section III describes the inspection setup and data sets, Section IV details the deep-learning architecture, Section V describes the DA methods used, Section VI provides results and discussion and Section VII the conclusion.

II. Relevant Research

Outside of NDE, TL has found success in a broad range of applications such as multilingual text classification, WiFi-based localization, speech recognition across different speakers, object recognition across different cameras, human motion parsing from videos, facial recognition and 3D pose estimation [21], [23], [24]. A major reason for this widespread usage of TL in recent years is the availability of large, free to access, source domain data, such as ImageNet [25] and CIFAR-10 [26] for natural image classification, IMdB reviews [27] and WordNet [28] for natural language processing, and LibriSpeech [29] for English speech recognition. For NDE there is a small, but insufficient, amount of work towards creating an equivalent data set [30]. But where source data is available, promising results with TL for NDE have been found. For example, a database of NDE X-ray images [31] has been used to train a CNN for inclusion detection in composites and unsupervised (i.e. without labeled target data) DA using the Case Western Reserve University bearing data set has been used to train a CNN for bearing inspections across different rotation speeds and load conditions. However, for most NDE applications, a training set large enough to function as source data for deep-learning is not available. Shallow-learning methods (i.e. predicting on hand selected features) require much less training data than deep-learning and have been used in structural health monitoring to train a hidden Markov model with source and target data from different transducer placements [32] and a K-Nearest Neighbors (KNN) method used to detect defects with source and target data from different carbon fiber composite samples [33]. A KNN model has also been used for structural health monitoring of buildings from the first three natural frequencies trained on source data from an analytical beam-bending model [34].

To find the most effective DA methods for use with labeled target data, as used in the current paper, research was conducted into popular deep learning DA methods proposed in recent published papers. During initial testing, some of the methods [35], [36] were found to produce lower sizing accuracy than networks trained without any target data at all, and are not presented here. The authors' believe that the poor performance of these methods is largely due to the fact that they are optimized for the 'semi-supervised' case where there is both unlabeled and labeled target data. Research specifically into supervised DA (i.e. where all data is labeled) methods has attracted little recent attention as most modern DA applications are motivated by lack of labeled data [24]. The authors found only two recently published methods specifically designed for supervised DA. These are *Regression and Contrastive Semantic Alignment* (*RCSA*) and *Adversarial*. *RCSA* uses an extra loss function to encourage proximity in the embedding space (the output of the convolutional layers) for data of the same label [37] while *Adversarial* optimally confuses a domain classifier to force the embedding space to be domain independent [38], [39]. These two DA methods are presented in the current paper along with a simpler DA approach, *MixedSet*, where training is performed with a mixed experimental/simulated set with sample weightings used to make up for the lack of experimental

data. As noted in [40] most DA research has focused on 'classification' tasks where the desired parameter is a discrete label. *RCSA* and *Adversarial* as originally presented in [37], [39] are consistent with this observation as they do not function with continuous labels. Because of this they have been adapted for the regression setting in this work; this is explained further in Section V. To the author's knowledge the only prior work in using simulated NDE data as a source domain for domain adapted deep-learning is [41] in which phased array data generated using a finite element model is used as source data to locate and size defects in an aluminum block. The authors of [41] use a basic DA approach in which they train on simulated, then experimental data. This method is similar to *MixedSet* in terms of its effect on the network.

III. INSPECTION SETUP AND DATA SET CREATION

This section describes the experimental acquisition and simulation of Plane Wave Capture (PWC) data and how it is imaged. The reader is directed to [20] for further detail. An outline of the resulting data set's size and parameter space is also given in this section.

A. Inspection Setup, Imaging and Simulation

Detecting and sizing cracks in the pipe wall is a major objective in inline-pipe inspection. These are usually caused by manufacturing faults such as weld toe cracks or in-service mechanisms, such as stress corrosion cracking, and most commonly occur at the outer surface of the pipe. Accurately sizing these surface-breaking cracks, once detected, is the objective of this work. As access to an oil pipeline was not available for this work, a representative inspection set up is used. As shown in Fig. 1a, an Imasonic (Voray-sur-l'Ognon, France) 5MHz, 0.3mm pitch, 40 element phased array in immersion is used to induce shear plane waves in 10mm 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 PWC data. Data is collected with the array positioned on both sides of the defect to mimic acquisition from a pair of arrays within the circumferential ring of arrays used on the pig. Each array fires

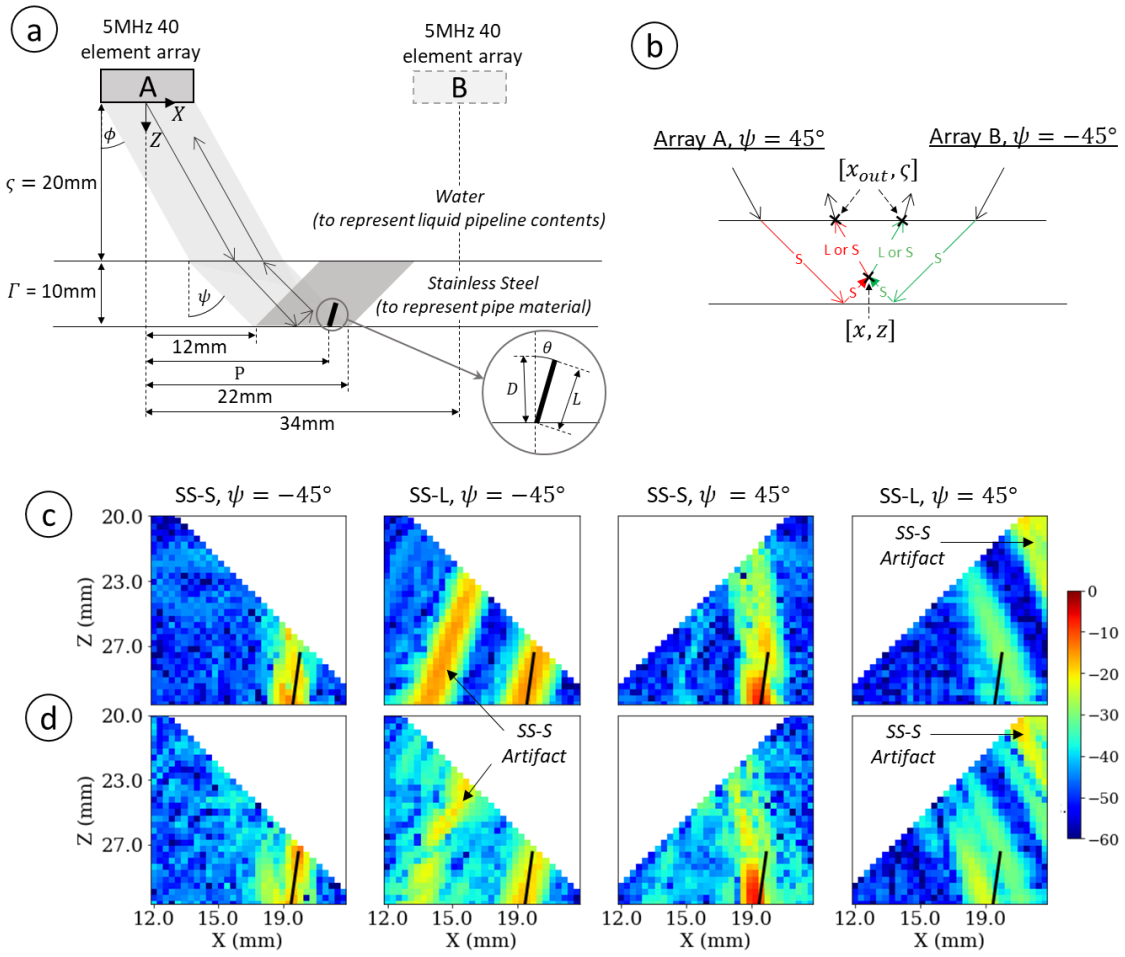

Fig. 1. a) A diagram of the inspection scenario using a plane wave at angle ψ to the vertical transmitted in the sample with a standoff and thickness of ς and Γ where L, θ and P represent the crack length, angle and position respectively, b) all half-skip shear (S) and longitudinal (L) mode ray-paths 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 [20].

a vertical 0° wave, which is used only to calculate standoff (ς) and pipe wall thickness (Γ). An angled wave at $\pm 19°$ inducing $\pm 45°$ shear plane waves in the sample is used to size the defects.

PWC data is focused on reception to create images, with the overall process referred to as Plane Wave Imaging (PWI) [42]. When different ray-paths are considered, these images are named 'views' and are categorized by the modality(s) of waves in their transmit and receive legs (L for longitudinal and S for shear). As half-skip modes have been shown to produce the strongest responses for surface-breaking defects [43] the SS-S and SS-L (Fig. 1b) views are used throughout. Each defect is imaged by an array on either side of it, ultimately producing four PWI images per defect. The region of interest is 12-22 mm from the array centre in the X-direction, and the full 10 mm of plate thickness in the Z-direction. A resolution of half a wavelength is used for imaging to minimize the data volume while preserving all information above the diffraction limit. This means that each image set input to the network is of size 32x32x4.

The simulated image sets are created by a hybrid Finite Element (FE) and ray-based model. As there is minimal transmission through the experimental defects they are modelled as rectangular, 0.3mm wide, perfect reflectors. Local FE analysis is used to calculate the response of a defect to a uni-modal plane wave. The FE model outputs scattering matrices [44] for each relevant length, L, and orientation, θ, of defect. These scattering matrices are then input into an analytical ray-based model [45] to form PWC data for each combination of L, θ, and horizontal position, P. The structural and grain noise is included by summation with one of 36 experimental, defect-free PWC sets [46]. The PWC data is then filtered to remove data outside the frequency range of the transducer using a Gaussian filter centered at 5 MHz with a -40 dB half width of 4.5 MHz before, finally, it is imaged to form the four relevant PWI images. An example set of experimental and simulated images is given in Fig. 1c,d.

B. Data Set Summary

While the target is the extent of the defect perpendicular to the surface, $D = L\cos\theta$, the parameter space of defects considered is defined by L, P and θ. All experimental defects used are 0.3 mm wide notches on the lower surface of the stainless-steel plate, manufactured using Electrical Discharge Machining (EDM). As described in Table I, the experimental data is from 1-5 mm in L, -20° to 20° in θ and 13-21mm in P. Negative θ data is obtained by positioning the array on the other side of the defect and variation in P achieved by moving the array relative to the defect. With negative and positive θ considered separately, this results in a total of $N_\theta \times N_L \times N_P = 11 \times 5 \times 27 = 1485$ image sets from the 30 manufactured defects. N_x indicates the number of possible values of x.

To ensure the trained network functions across the full parameter space the simulated set covers lengths and angles beyond that of the experimental set. This is described in Table II. Lengths above 5mm have not been considered as they are larger than the imaging domain, hence will be sized at 5mm. As critical crack depth is usually considered to be ~4 mm, sizing defects with $D > 5$ mm to $\hat{D} = 5$ mm is not an issue for this inspection. The simulated data totals 16,875 image sets. For machine learning purposes the data sets are split into a further four categories:

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

Simulated, validation: 15% (2,532) of simulated data used to qualitatively ensure the network is not overfitting to the training set.

Experimental, training: 3% to 49% (54-729) of experimental data used as 'target' in the DA methods to iteratively update the weights and biases of the network. The size of this set varies to investigate the effect on network accuracy.

Experimental, testing: 51% (756) of experimental data used to measure the sizing accuracy of the resulting network on previously unseen data.

Tables I and II describes how these sets are formed from the available experimental and simulated data. The split of data used for testing is fixed for all methods, meaning that this data is never used by any method during the training stage. As this work is motivated by creating an accurate sizing network with a minimum amount of NDE samples, the effect of the amount

TABLE I
Experimental Data Set Summary

Crack Angle, θ (°)	*Crack Length, L (mm)* 1	2	3	4	5
0	**Tr_2**	Test	**Tr_6**	Test	**Tr_3**
±2	Test	**Tr_{11}**	Test	**Tr_{10}**	Test
±5	Test	Test	Test	Test	**Tr_7**
±8	**Tr_8**	**Tr_{14}**	**Tr_5**	**Tr_{15}**	Test
±15	Test	**Tr_{12}**	Test	**Tr_{13}**	Test
±20	**Tr_4**	Test	**Tr_9**	Test	**Tr_1**

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

All Training = $N_{\theta,L} \times N_P$ = 27×27 = **729 image sets**

Testing = $N_{\theta,L} \times N_P$ = 28×27 = **756 image sets**

Tr_i represents the experimental training data for DA methods. When testing the effect of training set size these are iteratively combined together in ascending order of i.

$N_{x,y}$ indicates the number of possible pairs of values x and y.

TABLE II
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**

of experimental training data is explored. This requires a way of systematically increasing the size of the experimental training set in a way that optimally covers the parameter space. To achieve this, the 5x6 parameter space of lengths and angles is considered as a Cartesian grid of potential data points with axes normalised to span the range [0,1]. The first training point is added at (1,1). Additional training data points are progressively added to the vacant sites in the grid, with each new training data point added at the vacant site that has the maximum Euclidean distance to the nearest existing training data point in the normalised axes. This method is referred to as 'uniform sampling' in the current paper. The resulting sampling regime is given in Table I where Tr_i relates to i_{th} point added. The remaining 15 points are used as the test set. This method has the added benefit of ensuring that all data relating to any given defect is placed in either the training or test set, and cannot be spread across both. Because of this, any test set accuracy gained from the DA methods should generalize across the $\{L, \theta\}$ space and is not due to parameters covered by the experimental training data.

IV. NETWORK ARCHITECTURE

As in [20] the CNN architecture used here is loosely based on image recognition architectures such as AlexNet and VGG-19 due to their widespread success in both classification and regression tasks. An off-the-shelf architecture is not optimal due to the difference in input size and the dissimilarity between NDE data and natural images. As illustrated in Fig. 2a, the network's input is made up of the four 32×32 PWI images stacked in a third dimension, akin to how natural image CNNs treat red, green and blue channels. The network is made up of repeating blocks of convolution and max pooling layers for feature extraction, followed by a pair of fully connected layers for regression, with ReLU activation used throughout. Hyperparameters are set by testing networks with varying depth, number of filters, size of filters and number of neurons in the dense layers. Adding complexity to the network increased sizing accuracy but with diminishing returns for very large networks. The design illustrated in Fig. 2a is set at the point where adding further complexity only gives marginal accuracy gains. Further detail on the design process for this architecture can be found in [20]. Small architecture changes have been made between the two separate networks defined in [20] (that predicted L and θ individually) and the current paper where only a single network is required to predict D. The single network used here matches the structure of the L prediction network in [20] other than an increase in dropout rate from 0.1 to 0.3 which results in ~4% better prediction accuracy on the validation set at the cost of needing ~200 more epochs to reach convergence. The *Adversarial* DA method requires an additional domain classifier network, which is illustrated in Fig. 2b and comprises a single hidden layer of 128 neurons. This design was also obtained by adding layers until accuracy improvement was minimal. The purpose of the domain classifier is explained further in Section V.E.

Training the sizing network with all methods presented in this paper is achieved using the state-of-the-art Adam optimizer [47]. A learning rate of 1×10^{-3} is used unless otherwise stated in Section V. This value is used as increasing it created instabilities during training and decreasing it did not improve the performance of the converged network. A mini-batch size

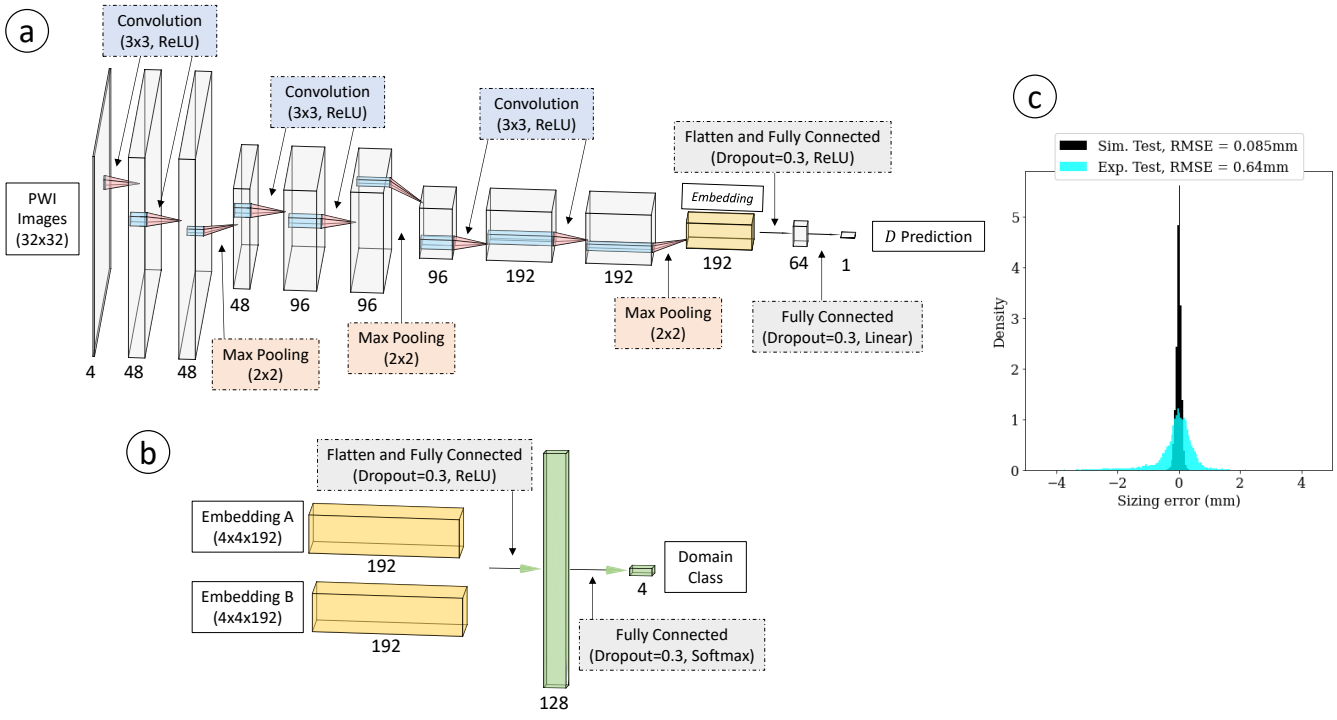

Fig. 2. Illustrations of a) the sizing CNN and b) the domain classifier used by *Adversarial* and c) the aggregated sizing error results for 20 initialisations of *SimOnly* applied to the experimental and simulated test sets.

of 64 is used unless the training set size is less than 64, in which case the entire set is processed at once. This is the case for *ExpOnly*, *RCSA* and *Adversarial* when the experimental training set contains only one defect. The number of epochs the network is trained for varies for each method and has been set to ensure convergence of the validation loss. Experimental and simulated D sizing errors for this network, trained only with simulated data, are illustrated in Fig. 2c. While a RMSE of 0.64 mm is significantly better than the performance of 6dB drop on this data set [20], the ~7.5 times lower simulated RMSE again motivates the need for DA when training using primarily simulated data. Example graphs of losses throughout training are given in the supplementary material.

V. DOMAIN ADAPTATION AND BASELINE METHODOLOGIES

This section describes the two baseline cases and three DA methodologies compared in this paper. As described in Section 2 these are the only DA methods the authors found in the literature specifically designed for the supervised setting. The baseline approaches train the CNN on the available data sets in isolation while the DA methodologies make use of both sets. The simplest of the DA approaches (*MixedSet*) trains on a mixture of the weighted experimental and simulated data while *RCSA* and *Adversarial* use different mechanisms to find an embedding space where the distributions of the data sets appear similar. As the output of the convolutional layers is the best approximation to the 'features' that the network is using to determine its final output [48], [49], this is selected as the embedding space.

A. Simulated Data Only (SimOnly)

This method makes no use of experimental data, except for testing. The network is trained with only the simulated training set, for 600 epochs, using Mean Square Error (MSE) as the loss function ($\mathcal{L}_R^s$).

B. Experimental Data Only (ExpOnly)

This method makes no use of simulated data, training the network with only the experimental training set, for 600 epochs, using MSE as the loss function ($\mathcal{L}_R^e$).

C. Mixture of Experimental and Simulata Data (MixedSet)

The training set for *MixedSet* is formed by shuffling together the M experimental and N simulated training image sets. The experimental data's contribution to the loss function is weighted by $\frac{N+M}{2M}$ and the simulated by $\frac{N+M}{2N}$ to ensure the large size of the simulated set does not swamp the effect of the experimental data [50]. The sizing network is trained on the combined set, using MSE as the loss function ($\mathcal{L}_R^{e,s}$), for 600 epochs

D. Regression and Contrastive Semantic Alignment (RCSA) [37]

RCSA combines the standard 'Regression' loss (MSE in this paper) with a 'Contrastive Semantic Alignment' loss that aims to force data with the same label (equivalent to the value of D in this paper) to be close in the embedding space, regardless of the domain. If this is achieved effectively it ensures that the features used by the fully connected layers to predict D, are domain independent. This means prediction accuracy learnt from simulated data should generalize well to experimental data, even if the particular $\{L, \theta\}$ combination tested was not present in the experimental training set.

RCSA functions by training a pair of networks with shared weights, one of which takes source domain data and the other target domain data. The distance metric used to define nearness in the embedding space must be selected. For this paper this has been set as the mean L_1 distance as lower orders of L_n caused instabilities in training and higher orders produced worse results. The Contrastive Semantic Alignment (CSA) loss was originally presented in [37] for classification of data with discrete labels where it is logical to cluster the same-label data into groups. Because of this, the CSA loss is formulated in [37] by penalizing distance between samples with the same label and rewarding distance between samples with different labels. To facilitate regression, it is more logical to have embedding distance be proportional to label difference. To this end the loss has been reformulated in this paper to encourage the distance between samples in the embedding space to scale with absolute difference in D. The L_1 norm is chosen to define the embedding space distance as it usually performs better than higher order norms for high-dimensional data [51]. The new CSA loss ($\mathcal{L}_{CSA}$) is therefore described by

$$\mathcal{L}_{CSA} = \frac{1}{M}\sum_{i=1}^{M}\left\{|D_i^s - D_i^e| - \frac{\sum_{j=1}^{\kappa}\left|E_{i,j}^s - E_{i,j}^e\right|}{\kappa}\right\} \quad (1)$$

where M is the size of the training set, D_i^s and D_i^e the simulated and experimental crack depths of the i_{th} image set, E_i^s and E_i^e the simulated and experimental embedding activations and κ the dimensionality of the embedding ($\kappa = 4 \times 4 \times 192 = 3072$ in this paper). The full *RCSA* loss ($\mathcal{L}_{RCSA}$) is given by

$$\mathcal{L}_{RCSA} = \mathcal{L}_R^s + \mathcal{L}_R^e + \alpha\mathcal{L}_{CSA} \quad (2)$$

where $\mathcal{L}_R^s$ and $\mathcal{L}_R^e$ are the regression losses (i.e. MSE) for the simulated and experimental data respectively and α is a tunable parameter that adjusts the relative importance of $\mathcal{L}_{CSA}$. The performance of the resulting network was found to be insensitive to the choice of α for the values tested (between 0.05 and 20) so, for simplicity, it is set to 1 in this work.

The training set for this method is formed by randomly pairing the experimental data with a sample of the simulated data meaning that M is equal to the size of the experimental training set. Both the pairings and the simulated data chosen are shuffled every 5 epochs to stop the network overfitting to any particular combination/subset. Training instabilities due to this overfitting occurred without implementing shuffling, but the resulting validation set accuracy was found to be insensitive to the choice of the frequency of shuffling provided it was <100 epochs. The network is trained for 5,000 epochs in total. Many more epochs are required to achieve convergence than for

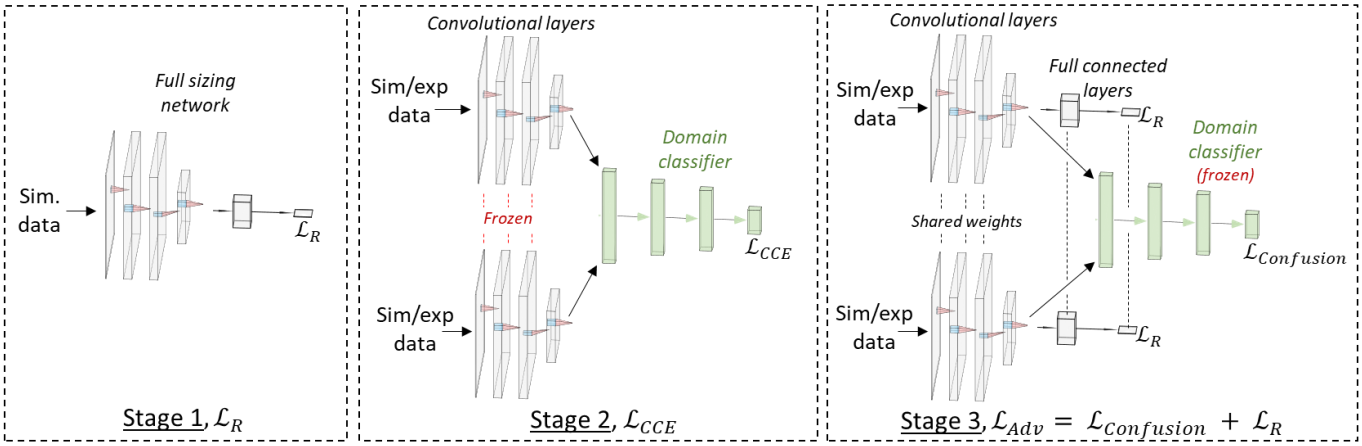

Fig. 3. An illustration of the three stages of training used in the *Adversarial* DA method.

SimOnly as each epoch only contains a small subset of the simulated data and an even smaller subset of all possible pairings.

E. Adversarial Domain Classifier (Adversarial) [38], [39]

A potentially impactful issue for *RCSA* is that in very high dimensional space, conventional distance metrics find most points to be equally far away from each other; this is a product of the 'curse of dimensionality' [52]. Adversarial DA bypasses the problem of finding a useful distance metric by training a separate neural network which aims to infer the domain of the data from embedding space activations (i.e. a domain discriminator). Once this is achieved, domain independent embeddings are achieved by maximally confusing the domain classifier.

As stated in [39], training a two-class domain discriminator with very little target data is difficult. The task is made easier by distinguishing between four cases: 1) Same label, same domain; 2) different label, same domain; 3) same label, different domain; and 4) different label, different domain). This approach does not have a natural reformulation for regression as the definition of 'same' and 'different' labels for continuous values is not clear. The equivalent proposed here is to say that if $|D^s - D^e| \leq \kappa$ then the labels are the same, where κ is a tolerance that depends on the application and availability of data. Here κ =1 mm is used, as this is the smallest value that can form 'same label, same domain' cases for the experimental data used in this paper.

The training process for *Adversarial* can be broken into three stages. These are illustrated in Fig. 3 and described in the following:

1. Train the sizing network with only the simulated data, minimizing MSE. As with the baseline methods, this is run for 600 epochs.
2. Form a weight shared pair of the convolutional blocks from stage 1. These convolutional blocks output into a domain classifier to predict which of the four groups a pair of data belong in. The architecture for the classifier is shown in Fig. 2b. This classifier is trained by freezing the weights and biases of the convolutional layers and minimizing the Categorical Cross Entropy ($\mathcal{L}_{CCE}$) which is described by

$$\mathcal{L}_{CCE} = -\frac{1}{M}\sum_{i=1}^{M} y_{i,1}\log\hat{y}_{i,1} + y_{i,2}\log\hat{y}_{i,2} + y_{i,3}\log\hat{y}_{i,3} + y_{i,4}\log\hat{y}_{i,4} \quad (3)$$

where $y_{i,j}$ is the binary class label for the i_{th} image set and j_{th} class and $\hat{y}_{i,j}$ the output of the domain classifier. This is run for 2400 epochs.

3. Both the convolutional and dense layers of the sizing network are trained whilst confusing the domain classifier with the weights and biases of the domain classifier frozen. The confusion loss ($\mathcal{L}_{Confusion}$)

$$\mathcal{L}_{Confusion} = -\frac{1}{M}\sum_{i=1}^{M} y_{i,1}\log\hat{y}_{i,3} + y_{i,2}\log\hat{y}_{i,4} + y_{i,3}\log\hat{y}_{i,1} + y_{i,4}\log\hat{y}_{i,2} \quad (4)$$

means that any changes made to the convolutional layers must maintain the domain classifiers label prediction accuracy whilst decreasing its domain prediction accuracy. The full adversarial loss ($\mathcal{L}_{Adv}$) is a trade-off between accurate sizing and domain independent embeddings and is defined by

$$\mathcal{L}_{Adv} = \mathcal{L}_R^s + \mathcal{L}_R^e + \beta\mathcal{L}_{Confusion} \quad (5)$$

where β is a tunable parameter that adjusts the relative importance of $\mathcal{L}_{Confusion}$. The performance of the resulting network was found to be insensitive to the choice of β for the values tested (between 0.05 and 20) so, for simplicity, it is set to 1 in this work. This is run for 2400 epochs.

The training set for stages two and three are formed in a similar fashion to *RCSA*, with pairs of experimental and simulated data. However, while for *RCSA* all data pairs are from different domains, for *Adversarial* some must be from the same domain so after pairing the sets they are shuffled across the domains. As with *RCSA* this pairing and shuffling is redone each 5 epochs. Learning rate for stage 3 is reduced to

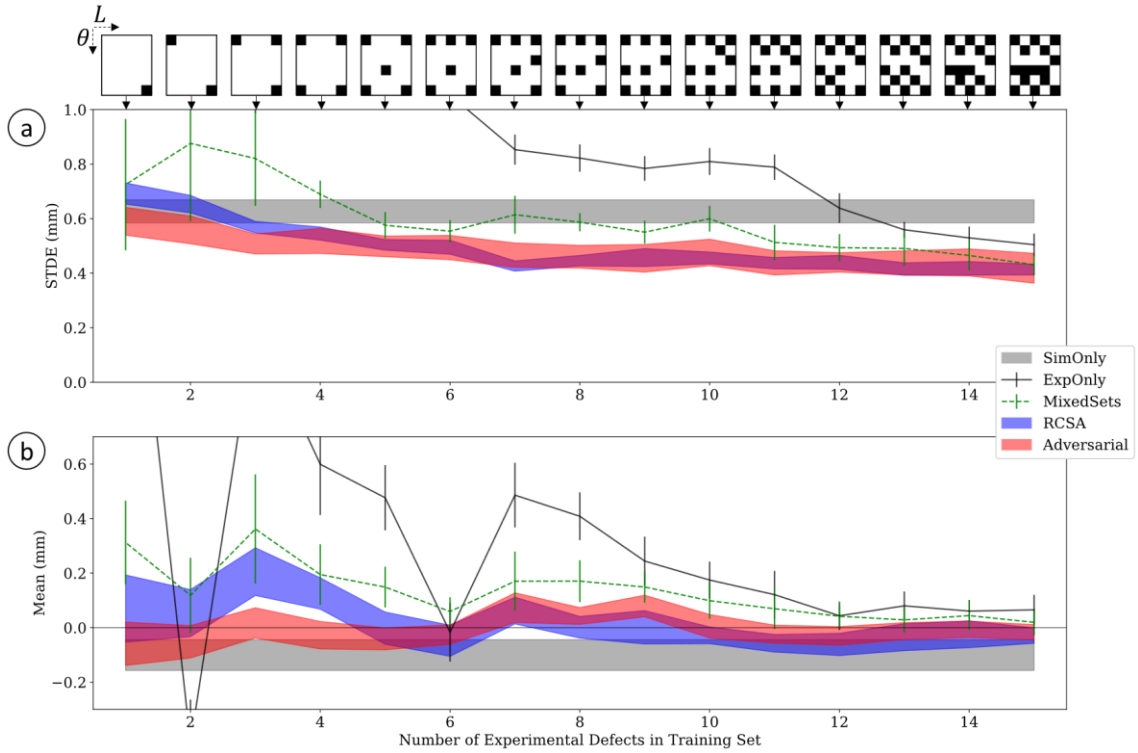

Fig. 4. a) The standard deviation in error (STDE) and b) the mean of the sizing error for the experimental test set across varying sizes of experimental training set. The error bars represent ± standard deviation over 20 independent initialisations. The graphics above the plots represent the $\{L, \theta\}$ coverage of the experimental training set.

0.2×10^{-3} to avoid gradient 'explosion' instabilities during training.

VI. Results and Discussion

The success of both the baseline and DA methods is measured by the sizing accuracy of the resulting networks on the unseen experimental test set. The mean error, and standard deviation of error (STDE) for varying experimental training set sizes is given in Fig. 4. The graphics at the top of Fig. 4 represent the $\{L, \theta\}$ space covered by the experimental training set. As the final network is affected by the initialisation of the weights and the train/validation shuffles, every point has error bars representing ± one standard deviation, based on results from 20 initialisations. For *SimOnly*, *RCSA* and *Adversarial* these error bars are shown as variable width lines for visual clarity. *SimOnly* produces networks with a STDE of 0.63 ± 0.04mm and a small negative mean of -0.10 ± 0.06mm, indicating a slight bias towards undersizing. As *SimOnly* makes no use of experimental training data these results are displayed as a constant grey band across Fig. 4.

The second baseline method, *ExpOnly*, is heavily reliant on having a large experimental training set. While it demonstrates greater accuracy than *SimOnly* with 13 or more defects in the training set, below this point, the STDE and mean increase quickly due to the network overfitting to the small set of training data. Overfitting, rather than more generalized learning, can be demonstrated by considering *ExpOnly* networks' performance on simulated data across the same $\{L, \theta\}$ space as the experimental test set. When trained with all 15 experimental defects, *ExpOnly* has a STDE of 1.02mm on simulated data, whereas the STDE of *SimOnly* on the experimental test set is 0.65mm. This asymmetry shows that while *SimOnly* can generalize reasonably well across the domain shift from simulated to experimental data, *ExpOnly* cannot do the reverse, and as a result, is unlikely to generalize well to even minor changes in inspection conditions (e.g. slight array movement, sound speed changes or crack roughness). This overfitting is likely caused by the significantly smaller training set available to *ExpOnly* compared to *SimOnly*.

MixedSet outperforms both of the baseline methods with 5 or more defects in the experimental training set but still suffers from inaccuracies due to overfitting when experimental data is scarce. The two other DA methods are given the same training data as *MixedSet* but perform better at all points. In terms of

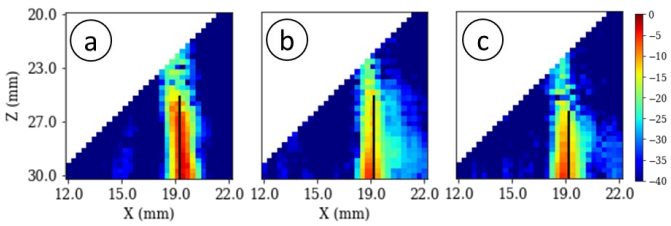

Fig. 5. a) Simulated and b) experimental SS-S PWI images for a defect with $P = 19mm, L = 5mm$ and $\theta = 8°$, and c) an experimental SS-S PWI image for a defect with with $P = 19mm, L = 4mm$ and $\theta = 0°$.

STDE, *RCSA* performs slightly worse than *SimOnly* with only one experimental training defect but at every other point outperforms both baseline methods and *MixedSet*. *Adversarial* gives the lowest STDE of all methods with 5 or less experimental training defects and has similar performance to *RCSA* above this point. The absolute mean for *RCSA* and *Adversarial* is negligible in most cases, becoming slightly larger for *RCSA* with low numbers of experimental training defects. This is likely due to uneven coverage of the $\{L, \theta\}$ parameter space.

It is clear from Fig. 4 that the two DA methods: *RCSA* and *Adversarial*, make better use of limited experimental data than *MixedSet*. This can be explained by their differing objectives. Rather than aiming for accurate experimental sizing directly, which is difficult with limited data, *RCSA* and *Adversarial* focus on extracting domain independent embeddings. This is an easier task to achieve with limited data. Also, if domain invariant embeddings are found between the $\{L, \theta\}$ examples in the experimental training set and the full simulated training set they are likely to generalize to all $\{L, \theta\}$ of interest as these are all present in the simulated training set.

The negative mean for *SimOnly* is caused by undersizing of 5 mm defects. This is because the far-field assumption of the simulation is inaccurate for defects larger than 4 mm as their tips enter the array's near-field. This inaccuracy is exampled in Fig. 5a where it can be seen that the simulation overestimates the amplitude of the tip reflection in comparison to the experimental data in Fig. 5b. *SimOnly* sizes the PWI data from the $D = 5$ mm defect shown in Fig. 5b to be of $\widehat{D} = 4.4$ mm which makes intuitive sense as, visually, the image appears closer to the experimental $D = 4$ mm defect in Fig. 5c (which SimOnly sizes as $\widehat{D} = 4.0$ mm) than the simulated $D = 5$ mm defect in Fig. 5a. This kind of simulation deficiency is a good example of the need for DA.

The effect of the position of the training data points in $\{L, \theta\}$ space is investigated by using *RCSA* and *Adversarial* with four experimental training defects but rather than using uniform sampling to optimally choose the $\{L, \theta\}$ combination, they are picked at random. The mean Euclidean distance between the training set examples in terms of normalised $\{L, \theta\}$ is used as an indication of how well sampled the parameter space is. The random selection of the training data points is repeated 8 times with different random number generator seeds (training shuffle number = 1-8). The results of this experiment are shown in Fig.

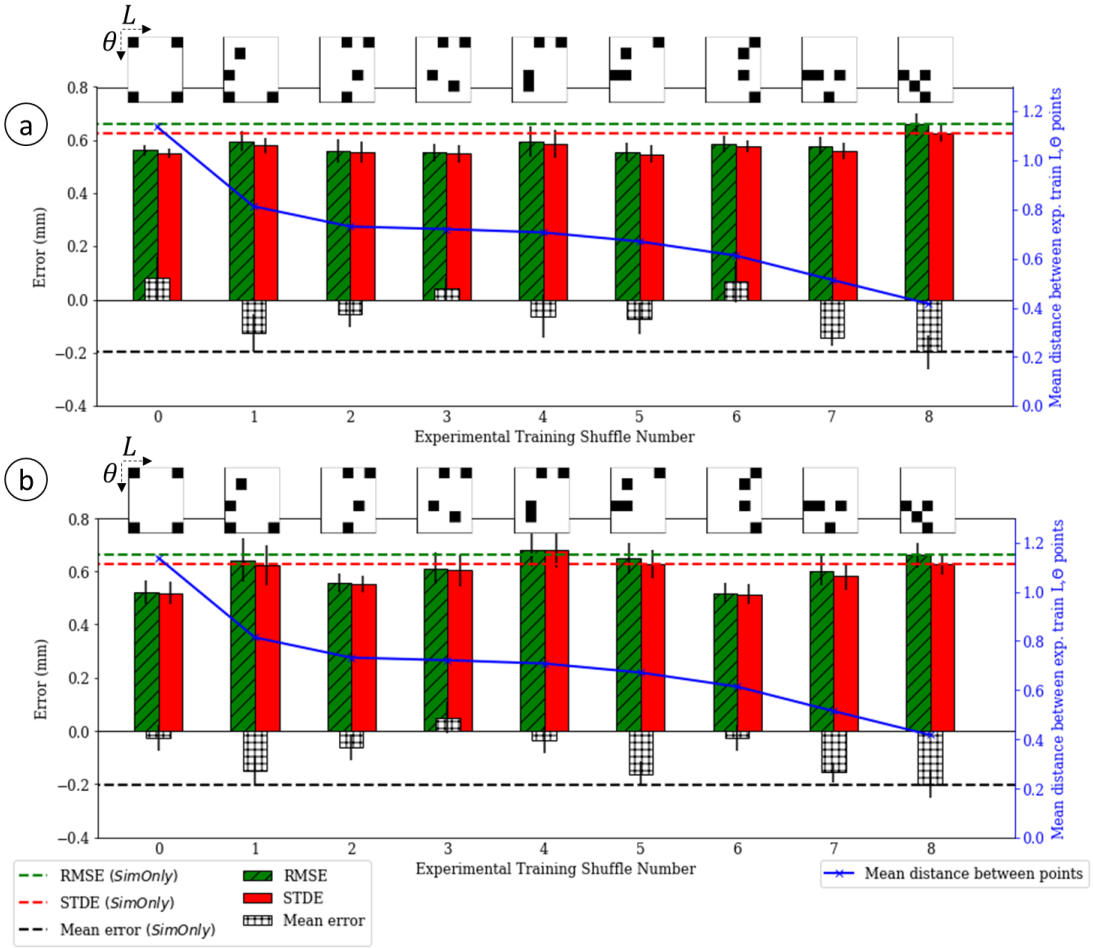

Fig. 6. The Root Mean Square Error (RMSE), mean error (μ), standard deviation of error (STDE) and mean Euclidean distance between points in normalised $\{L, \theta\}$ space for a) *RCSA* and b) *Adversarial* methods with varying choices of experimental training set. The graphics above the plots represent the $\{L, \theta\}$ coverage of the four experimental defects.

6. Root Mean Squared Error (RMSE) is reported alongside STDE and mean error to provide a single metric with which sizing accuracy can be easily compared across shuffle numbers.

In Fig. 6, the results are presented in order of decreasing mean distance between points in $\{L, \theta\}$ space as a measure of parameter space coverage. A reduction in parameter space coverage might intuitively be expected to lead to increased RMSE, but no correlation is discernible in Fig. 6. However, both methods produce the lowest errors in the uniformly sampled case (training shuffle number = 0), compared to other possibilities. For both methods, the most poorly sampled case (training shuffle number = 8) offered no accuracy increase over *SimOnly*. This demonstrates the importance of sampling the defect's parameter space as evenly as possible with the available experimental training data.

VII. Conclusions

This paper has demonstrated the ability of modern DA methods to improve the accuracy of deep networks for defect sizing, trained on simulated data, with even a very limited amount of experimental data. The key metrics for comparison of the methods considered are illustrated in Fig. 7. *Adversarial* and *RCSA* produced the most accurate networks for all sizes of experimental training set with *Adversarial* outperforming *RCSA* with less than 6 experimental training defects. With only 4 experimental defects *RCSA* and *Adversarial* reduced STDE on the experimental test set by 13% and 17% respectively, compared to *SimOnly*. However, *RCSA* is the easier method to implement as it only introduces one extra tunable parameter (loss function scaling factor, α) while *Adversarial* requires tuning of β, design of the architecture for the domain classifier, and takes almost ~10 times longer to train than *RCSA* when the experimental training set is small. The success of both modern DA methods was shown to be sensitive to coverage of the $\{L, \theta\}$ parameter space by the experimental training set. The results of this paper suggest that uniform sampling, starting at the corners of the parameter space, is an effective way of designing a small experimental training set. Optimal sampling for higher dimensional parameter spaces needs further investigation.

Future research should be carried out to investigate the impact of larger gaps in the distributions of source and target domains. For example, testing if ultrasonic data from a different inspection or even natural images would be useful source domains. The possibility for using NDE specific data augmentation alongside the DA methods presented here to further increase the usefulness of small pools of experimental data should also be investigated. Another major improvement would be to use probabilistic methods to add values of uncertainty to the predictions of the deep learning network. The modern DA methods presented in this paper are shown to be successful for improving the accuracy of deep learning for in-line pipe inspection and, as they are agnostic to the structure of the data, are expected to be applicable to other NDE inspections and modalities.

References

[1] A. S. Lundervold and A. Lundervold, "An overview of deep learning in medical imaging focusing on MRI," *Z. Med. Phys.*, vol. 29, no. 2, pp. 102–127, 2019.

[2] D.-P. Marija Bertović, "Human Factors in Non-Destructive Testing (NDT): Risks and Challenges of Mechanised NDT," 2016. Accessed: Jun. 01, 2020. [Online]. Available: www.bam.de.

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

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

[5] 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," *J. Manuf. Process.*, vol. 52, pp. 26–34, Apr. 2020, doi: 10.1016/j.jmapro.2020.01.047.

[6] 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," *J. Civ. Struct. Heal. Monit.*, pp. 1–15, Mar. 2020, doi: 10.1007/s13349-020-00391-7.

[7] A. Bernieri, L. Ferrigno, M. Laracca, and M.

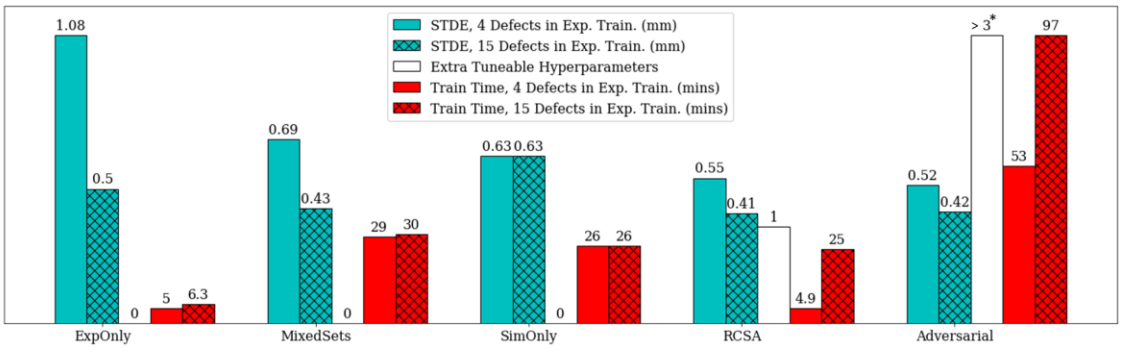

Fig. 7. Summary of the key properties of the methods investigated in this paper. *This includes full design of the domain classifier architecture.

Molinara, “Crack shape reconstruction in Eddy current testing using machine learning systems for regression,” *IEEE Trans. Instrum. Meas.*, vol. 57, no. 9, pp. 1958–1968, 2008, doi: 10.1109/TIM.2008.919011.

[8] 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.

[9] 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.

[10] I. Virkkunen, T. Koskinen, O. Jessen-Juhler, and J. Rinta-Aho, “Augmented ultrasonic data for machine learning,” *J. Nondestruct. Eval.*, vol. 40, no. 1, pp. 1–11, 2021.

[11] J. B. Harley and D. Sparkman, “Machine learning and NDE: Past, present, and future,” in *AIP Conference Proceedings*, 2019, vol. 2102, no. 1, p. 90001.

[12] J. Feng, F. Li, S. Lu, J. Liu, and D. Ma, “Injurious or noninjurious defect identification from MFL images in pipeline inspection using convolutional neural network,” *IEEE Trans. Instrum. Meas.*, vol. 66, no. 7, pp. 1883–1892, 2017.

[13] I. Virkkunen, T. Koskinen, O. Jessen-Juhler, and J. Rinta-Aho, “Augmented Ultrasonic Data for Machine Learning,” *arXiv Prepr. arXiv1903.11399*, 2019.

[14] J. C. Aldrin and D. S. Forsyth, “Demonstration of using signal feature extraction and deep learning neural networks with ultrasonic data for detecting challenging discontinuities in composite panels,” in *AIP Conference Proceedings*, May 2019, vol. 2102, no. 1, doi: 10.1063/1.5099716.

[15] H. wei Huang, Q. tong Li, and D. ming Zhang, “Deep learning based image recognition for crack and leakage defects of metro shield tunnel,” *Tunn. Undergr. Sp. Technol.*, vol. 77, pp. 166–176, Jul. 2018, doi: 10.1016/j.tust.2018.04.002.

[16] L. Perez and J. Wang, “The effectiveness of data augmentation in image classification using deep learning,” *arXiv Prepr. arXiv1712.04621*, 2017.

[17] O. Siljama, T. Koskinen, O. Jessen-Juhler, and I. Virkkunen, “Automated Flaw Detection in Multi-channel Phased Array Ultrasonic Data Using Machine Learning,” *J. Nondestruct. Eval.*, vol. 40, no. 3, pp. 1–13, 2021.

[18] D. Mery, “Aluminum Casting Inspection using Deep Object Detection Methods and Simulated Ellipsoidal Defects,” *Mach. Vis. Appl.*, vol. 32, no. 3, pp. 1–16, 2021.

[19] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[20] R. J. Pyle, R. L. T. Bevan, R. R. Hughes, R. K. Rachev, A. A. S. Ali, and P. D. Wilcox, “Deep Learning for Ultrasonic Crack Characterization in NDE,” *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, 2020.

[21] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, 2009.

[22] Google, “Machine Learning Glossary | Google Developers,” *Machine Learning Crash Course*. https://developers.google.com/machine-learning/glossary (accessed Jun. 10, 2020).

[23] M. Wang and W. Deng, “Deep visual domain adaptation: A survey,” *Neurocomputing*, vol. 312, pp. 135–153, 2018.

[24] G. Csurka, “Domain adaptation for visual applications: A comprehensive survey,” *arXiv Prepr. arXiv1702.05374*, 2017.

[25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*, 2009, pp. 248–255.

[26] A. Krizhevsky and G. Hinton, “Learning multiple layers of features from tiny images,” 2009.

[27] A. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, “Learning word vectors for sentiment analysis,” in *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, 2011, pp. 142–150.

[28] G. A. Miller, *WordNet: An electronic lexical database*. MIT press, 1998.

[29] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: an asr corpus based on public domain audio books,” in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2015, pp. 5206–5210.

[30] J. Ye and N. Toyama, “Benchmarking Deep Learning Models for Automatic Ultrasonic Imaging Inspection,” *IEEE Access*, vol. 9, pp. 36986–36994, 2021.

[31] D. Mery *et al.*, “GDXray: The database of X-ray images for nondestructive testing,” *J. Nondestruct. Eval.*, vol. 34, no. 4, p. 42, 2015.

[32] D. Chakraborty, N. Kovvali, B. Chakraborty, A. Papandreou-Suppappola, and A. Chattopadhyay, “Structural damage detection with insufficient data using transfer learning techniques,” in *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2011*, 2011, vol. 7981, p. 798147.

[33] P. Gardner *et al.*, “Machine learning at the interface of structural health monitoring and non-destructive evaluation,” *Philos. Trans. R. Soc. A*, vol. 378, no. 2182, p. 20190581, 2020.

[34] P. Gardner, X. Liu, and K. Worden, “On the application of domain adaptation in structural health monitoring,” *Mech. Syst. Signal Process.*, vol. 138, p. 106550, 2020.

[35] H. Daumé III, A. Kumar, and A. Saha, “Frustratingly easy semi-supervised domain adaptation,” in *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, 2010, pp. 53–59.

[36] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, “Domain adaptation via transfer component analysis,”

IEEE Trans. Neural Networks, vol. 22, no. 2, pp. 199–210, 2010.

[37] S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto, "Unified deep supervised domain adaptation and generalization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5715–5725.

[38] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7167–7176.

[39] S. Motiian, Q. Jones, S. Iranmanesh, and G. Doretto, "Few-shot adversarial domain adaptation," in *Advances in Neural Information Processing Systems*, 2017, pp. 6670–6680.

[40] A. Singh and S. Chakraborty, "Deep Domain Adaptation for Regression," in *Development and Analysis of Deep Learning Architectures*, W. Pedrycz and S.-M. Chen, Eds. Cham: Springer International Publishing, 2020, pp. 91–115.

[41] T. Latête, B. Gauthier, and P. Belanger, "Towards using convolutional neural network to locate, identify and size defects in phased array ultrasonic testing," *Ultrasonics*, vol. 115, p. 106436, 2021.

[42] 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.

[43] M. V Felice, A. Velichko, and P. D. Wilcox, "Accurate depth measurement of small surface-breaking cracks using an ultrasonic array post-processing technique," *Ndt E Int.*, vol. 68, pp. 105–112, 2014.

[44] 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.

[45] 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 Trans. Ultrason. Ferroelectr. Freq. Control*, pp. 1–1, Jan. 2020, doi: 10.1109/tuffc.2020.2969083.

[46] 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.

[47] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, Dec. 2015.

[48] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," *arXiv Prepr. arXiv1405.3531*, 2014.

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

[50] scikit-learn, "sklearn.utils.class_weight.compute_class_weight," 2020. https://scikit-learn.org/stable/modules/generated/sklearn.utils.class_weight.compute_class_weight.html (accessed May 06, 2021).

[51] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, "On the surprising behavior of distance metrics in high dimensional space," in *International conference on database theory*, 2001, pp. 420–434.

[52] M. Köppen, "The curse of dimensionality," in *5th Online World Conference on Soft Computing in Industrial Applications (WSC5)*, 2000, vol. 1, pp. 4–8.

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.

Rhodri L. T. Bevan received an M.Eng. degree in Civil Engineering in 2005, an M.Res. in Computational Engineering in 2006 and a Ph.D. from Swansea University (Swansea, UK) in 2011.

From 2011 to 2013 he was a Research Assistant in the Civil and Computational Engineering Centre at Swansea University developing high-performance finite element programs. From 2013 to 2016 he was a Research Assistant in Numeric Simulation and Optimisation in the Department of Aerospace Engineering at the University of Bristol. Since 2016, Dr. Bevan has been with the Department of Mechanical Engineering at the University of Bristol and his current research interests include machine learning for defect detection and characterization, linear and non-linear array imaging, data fusion and high-performance algorithm design.

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.

SUPPLEMENTARY MATERIAL

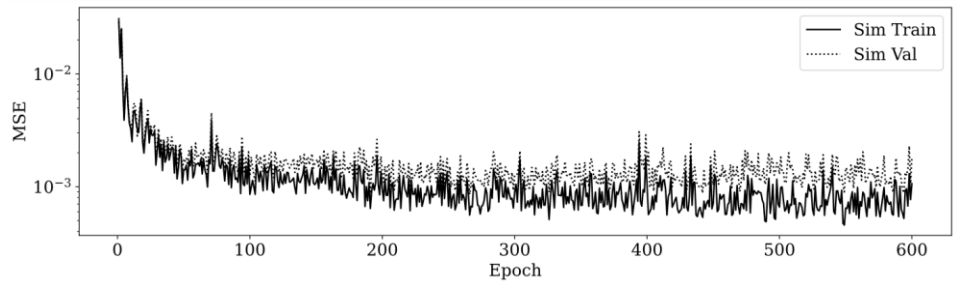

Fig. 8. Training and validation losses through training for *SimOnly*.

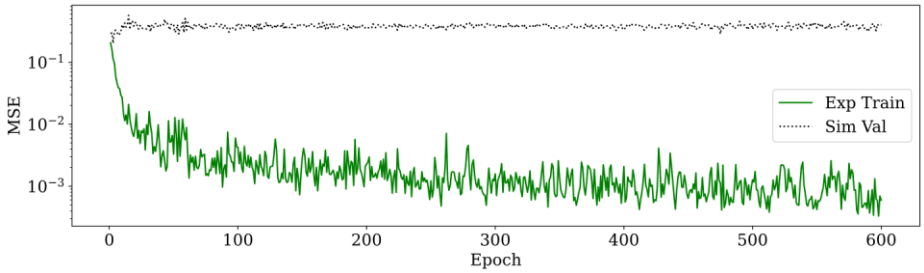

Fig. 9. Training and validation losses through training for *ExpOnly* with all 14 defects.

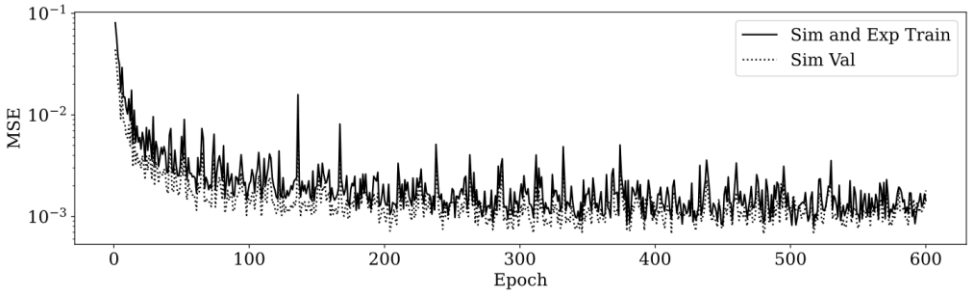

Fig. 10. Training and validation losses through training for *MixedSets* with all 14 defects.

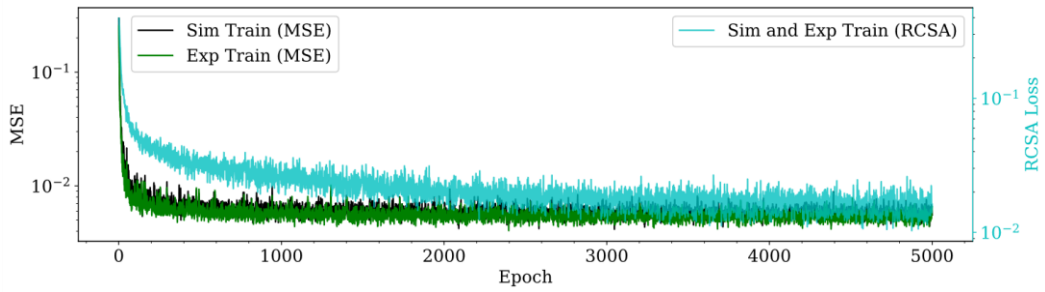

Fig. 11. Training and validation losses through training for *RCSA* with all 14 defects.

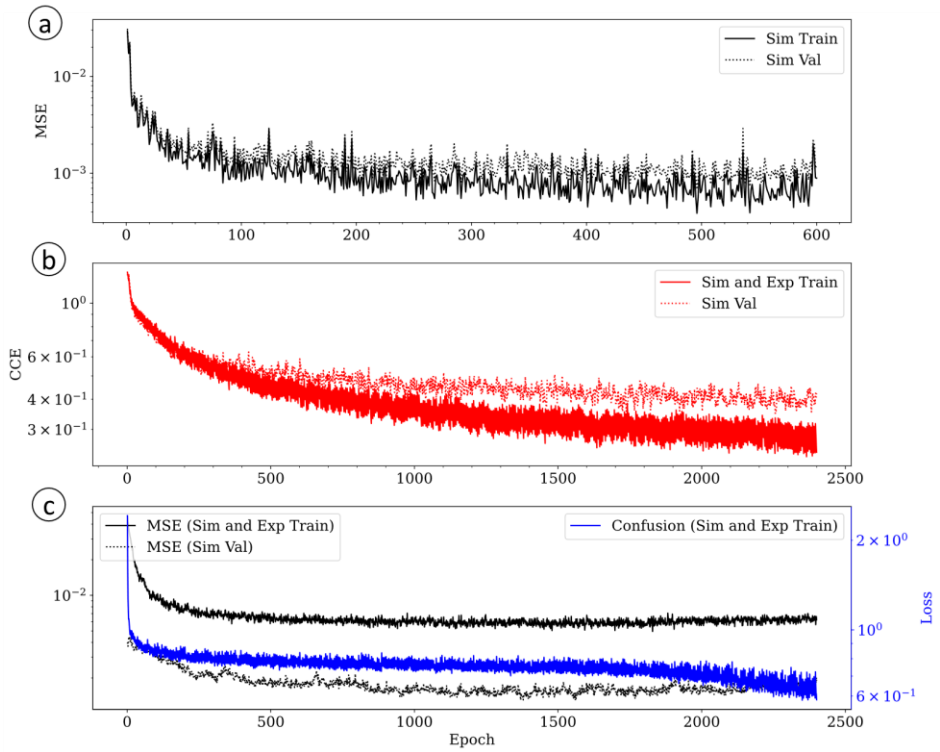

Fig. 12. Training and validation losses for *Adversarial,* with all 14 defects, through training for step a) training on simulated data b) training the domain classifier and c) training on simulated and experimental data while confusing the domain classifier.