ORIGINAL ARTICLE

Real-time super-resolution mapping of locally anisotropic grain orientations for ultrasonic non-destructive evaluation of crystalline material

Jonathan Singh[1] · Katherine Tant[1] · Andrew Curtis[2] · Anthony Mulholland[3]

Abstract

Estimating the spatially varying microstructures of heterogeneous and locally anisotropic media non-destructively is necessary for the accurate detection of flaws and reliable monitoring of manufacturing processes. Conventional algorithms used for solving this inverse problem come with significant computational cost, particularly in the case of high-dimensional, nonlinear tomographic problems, and are thus not suitable for near-real-time applications. In this paper, for the first time, we propose a framework which uses deep neural networks (DNNs) with full aperture, pitch-catch and pulse-echo transducer configurations, to reconstruct material maps of crystallographic orientation. We also present the first application of generative adversarial networks (GANs) to achieve super-resolution of ultrasonic tomographic images, providing a factor-four increase in image resolution and up to a 50% increase in structural similarity. The importance of including appropriate prior knowledge in the GAN training data set to increase inversion accuracy is demonstrated: known information about the material's structure should be represented in the training data. We show that after a computationally expensive training process, the DNNs and GANs can be used in less than 1 second (0.9 s on a standard desktop computer) to provide a high-resolution map of the material's grain orientations, addressing the challenge of significant computational cost faced by conventional tomography algorithms.

Keywords Deep neural networks · Generative adversarial networks · Ultrasound tomography · Anisotropy

1 Introduction

Ultrasonic non-destructive evaluation (NDE) is widely used across a number of industries including aerospace, nuclear and oil and gas. The technique involves the generation, transmission and reception of high-frequency mechanical waves through a component [11]. An image of the component's interior is then generated via post-processing of this data to aid in the detection of any internal defects [42]. Conventional ultrasonic imaging algorithms

within NDE typically assume that the material that is being inspected is isotropic and homogeneous. However, metals can develop locally anisotropic and heterogeneous microstructures, particularly when they are subjected to extreme thermal cycles, such as those present in welding and additive manufacturing processes [16, 51, 60]. Conventional ultrasonic imaging algorithms which assume homogeneity or isotropy can fail to focus the energy correctly in the image domain in such cases and are therefore unreliable [44, 55, 66]. Algorithms which incorporate *a priori* information about a material's spatially varying properties significantly improve the accuracy of defect characterisation [55].

In recent years, much effort has been expended on generating material property maps non-destructively using tomographic inversion, where material properties such as wave speed, or microstructural descriptors such as grain orientation, are estimated from the scattered wave field data recorded at the surface of an object. A wide range of

✉ Jonathan Singh
jonathan.singh@strath.ac.uk

1 Mathematics and Statistics, University of Strathclyde, Glasgow, UK

2 School of Geoscience, University of Edinburgh, Edinburgh, UK

3 Engineering Mathematics, University of Bristol, Bristol, UK

advanced tomographic algorithms are used across geophysics [1, 2, 12, 40, 58, 67, 69], bio-medicine [19] and NDE [14, 37, 55, 56]. A common approach is to use iterative methods to improve the fit of the measured data to forward modelled data which depend on an estimate of the material map. They sample potential material maps from some multi-dimensional parameter space, solve a forward problem for each new material property map and update the estimated map to improve the data fit [37]. In the case of probabilistic sampling frameworks (for example, those built around Markov chain Monte Carlo methods [56, 68]), there is the added benefit of extracting uncertainty information on the parameter estimates, facilitating valuable uncertainty quantification studies. Although these algorithms have demonstrated impressive results in reconstructing wave speed and grain orientation maps, they are computationally demanding, often requiring the storage of large sample sets and compute times of several hours to several weeks. This poses a problem for the NDE community, where there is an increasing demand for the monitoring of dynamical processes employed during manufacturing, for example, in welding and additive manufacturing processes [31, 32], and so it is desirable to carry out inspection in real time.

Machine learning shows strong potential to solve isotropic material characterisation inverse problems rapidly [20] and has comparable results to more computationally expensive algorithms such as Markov chain Monte Carlo methods [21]. Specifically, we focus on the use of deep neural networks (DNNs), which can approximate any nonlinear relationship between two parameter spaces, given a sufficiently large set of training data (pairs of dependent and corresponding independent parameters [9]). The training of a DNN is computationally expensive. However, the training process is only performed once prior to using a DNN, and a trained network can be used effectively in near-real time without the need for high-performance computing.

Inversion methods based on DNNs have become increasingly popular for tomographic imaging of isotropic material properties, particularly in geophysics [5, 8, 13, 20, 43] and bio-medicine [4, 64]. However, DNNs have not yet been implemented for tomographic reconstruction of anisotropic material properties. Although various deep learning algorithms have been used to solve inverse problems in NDE, for example, to predict material fatigue behaviour [3], to augment ultrasonic data [59] and for ultrasonic crack characterisation [47] and crack detection using image recognition [15, 29], the use of DNNs for tomography has yet to be explored in this context.

In addition to DNNs, generative adversarial networks (GANs) have more recently been applied to various computer vision tasks, including achieving super-resolution

with upscaling by up to a factor of four [41], colourisation [25], segmentation and labelling [30]. This family of algorithms has strong potential to improve image resolution and has been used increasingly in remote sensing [33] and X-ray tomography [65]; however, there has been no application of GANs in NDE to produce ultrasonic tomographic images. Achieving image super-resolution is a challenging task, and a range of algorithms have been employed to tackle this problem including interpolation-based methods [36], reconstruction-based methods [17] or learning-based methods [26]. Interpolation-based and reconstruction-based methods can suffer from accuracy shortcomings, particularly when the super-resolution scale factor increases, whereas learning-based methods such as GANs are increasingly used for their fast computation and good performance [63]. Therefore, we focus on such learning-based methods for our application.

The novel elements of this paper are: (1) the first DNN framework for rapid, nonlinear, two-dimensional tomography of heterogeneous and locally anisotropic materials and (2) the first use of GANs for processing ultrasound tomographic images in NDE. The data sets used for the tomographic inversion are the arrival times of ultrasonic waves which have been transmitted and received by an array of sensors on the exterior of the component. The examples shown are inspired by the NDE of polycrystalline materials, but the methodology should naturally extend to other domains, for example, imaging anisotropic fibrous tissue [22, 27] or the Earth's subsurface [70]. We compare the network's performance for a range of transducer configurations, model textures and different types of simulated ultrasonic testing data (i.e. we move beyond inverse crime scenarios). The novel GAN-based method for post-processing ultrasound tomographic images to achieve super-resolution with a fourfold upscaling factor is presented, achieving up to 50% improvement using structural similarity metrics. We define the term super-resolution in the context of image processing, as reconstructing images below the original lengthscale. This is different to an alternative definition often used in physical acoustics, which is to image below the wavelength in the data.

2 Method

We employ model-driven deep learning, where a large data set of simulated material maps and corresponding travel time measurements are used to train a DNN and hence solve the tomographic inverse problem. The forward modelling problem can be denoted as

$$f(\mathbf{m}, \mathbf{s}) = \mathbf{T_m}, \tag{1}$$

where f is a forward mechanical wave modelling operator, $\mathbf{m}$ is a material model, $\mathbf{s}$ contains the locations of the elements in the ultrasonic transducer array and $\mathbf{T_m}$ is the time-of-flight (ToF) matrix between every pair of array elements. Within each database used for network training, the transducer configuration $\mathbf{s}$ is fixed and therefore $\mathbf{s}$ is omitted in the notation for the ToF matrix $\mathbf{T_m}$. We use deep learning to obtain (or learn) an approximation of f^{-1}, which maps the measured data $\mathbf{T_m}$ to a material map $\mathbf{m}$ (i.e. DNN $\approx f^{-1}$). In this study, the training data consist of two-dimensional material models with spatially varying crystal orientations $\theta(x,y)$ and the travel time matrix $\mathbf{T_m}$ corresponding to each one. The target materials for characterisation are metals, which often exhibit correlated structures due to their manufacturing process (i.e. an interlocking crystalline texture). While Earp et al. [20] successfully use both normal and uniformly distributed random models without correlated structure in the training data, here we generate models in such a way that although the distribution of orientations is randomly assigned, the material still exhibits some structural correlation which well represents the microstructure of the material of interest. To achieve this, we examine that an initial random Voronoi tessellation [53] with 30 seeds (a set of two-dimensional Cartesian coordinates lying within the domain of interest) is computed and an orientation θ between $0°$ and $45°$ is randomly assigned to each of the 30 resulting Voronoi regions or cells (Fig. 1a). We consider only in-plane crystal rotation, and therefore, the orientation θ relates to the orientation of a slowness curve in each cell. This slowness curve plots the reciprocal of velocity in the crystal over a range of incident wave directions [56]. The material models used in the training data $\{\mathbf{m}_{16}, \mathbf{T_{m_{16}}}\}$ are generated by discretising the Voronoi tessellation into a regularly spaced 16×16 grid and smoothing with a Gaussian kernel (Fig. 1b; the subscript 16 denotes the model resolution). Gaussian smoothing is less likely to cause low-frequency artefacts compared to other methods such as a moving average approach, and convolutional neural layers have been proven to be effective for Gaussian denoising [39]. The smoothing simplifies the inverse problem such that only smooth models are inverted for. To demonstrate that this machine learning approach can be generalised for any locally anisotropic media, the longitudinal group slowness curve is obtained for an arbitrary anisotropic material with a cubic stiffness tensor, where $c_{11} = 256.45$ GPa, $c_{12} = 133.5$ GPa and $c_{44} = c_{12}$ and density $\rho = 7874$ kg m^{-3}. Three common configurations of ultrasonic transducer array locations $\mathbf{s}$ are considered: a full aperture coverage of 16 elements (4 on each face as shown in Fig. 1d), a two-sided aperture pitch-catch configuration with 16 transmitting elements at the top of the model, with the time of

flights measured at 16 receiving elements at the bottom of the model (Fig. 1e) and a one-sided aperture pulse-echo configuration, where 16 elements are positioned along the top face and the travel times of waves reflecting off the bottom face and returning to the transducers on the top face are measured (Fig. 1f). In real-world applications, often only the pulse-echo configuration is feasible due to access of the test sample, but to develop the algorithms, the availability of data from the full aperture to the pitch-catch to the pulse-echo arrangements is gradually decreased. The measured data are the time of flight (ToF) of each propagating wave between each pair of array elements, represented in a ToF matrix $\mathbf{T_{m_{16}}}$ shown in Fig. 1c.

2.1 Forward model approaches

Acquiring the data for training a DNN experimentally would be impractical due to the time and cost of obtaining the large amount of data that is required. So an efficient forward model is needed for computing the time-of-flight matrix $\mathbf{T_m}$ (Fig. 1d) corresponding to a grain orientation model $\mathbf{m}_{16}$ for each source–receiver pair. We take two approaches: a semi-analytic model using an anisotropic multi-stencil fast marching method (AMSFMM) algorithm from [56], denoted as f_{FMM}, and a finite element analysis (FEA) method, denoted as f_{FEA}. The AMSFMM incorporates the effects of ray bending due to variations in locally anisotropic grain orientations and models the travel-time field by solving the Eikonal equation using an upwind finite difference scheme [49, 54, 56]. This allows the calculation of the shortest travel time between transmitter and receiver locations, and the matrix $\mathbf{T_m}^{FMM}$ can be constructed (that is $\mathbf{T_m}^{FMM} = f_{FMM}(\mathbf{m}_{16})$). As wave reflections are not incorporated into the AMSFMM, a different approach is required for the pulse-echo transducer array configuration. In this case, the time of flight between the transmitter and receiver is calculated by the summation of the time of flight between the transmitter to all points along the back-wall and between the receiver and all points along the bottom face. The output of this summation is an array of travel times corresponding to all the reflection points along the bottom face, and the minimum value is taken to be the time of flight for the pulse-echo transducer array configuration. The FEA method incorporates more of the underlying physics in the model compared to AMSFMM, as it models full wave propagation including multiple scattering and diffraction. Following the approach of [56], to measure the ToF of the received waves from FEA generated data, an amplitude threshold is selected and the time for the recorded wave amplitude to reach this threshold is used as an element of the travel time matrix $\mathbf{T_m}^{FEA}$ (that is $\mathbf{T_m}^{FEA} = f_{FEA}(\mathbf{m}_{16})$). The FEA method is significantly

(a) Initial Voronoi Tesselation

(b) 16x16 Smoothed Model

(c) Travel Time Matrix

(d) Full Aperture Transducer Coverage

(e) Pitch-Catch Configuration

(f) Pulse-Echo Configuration

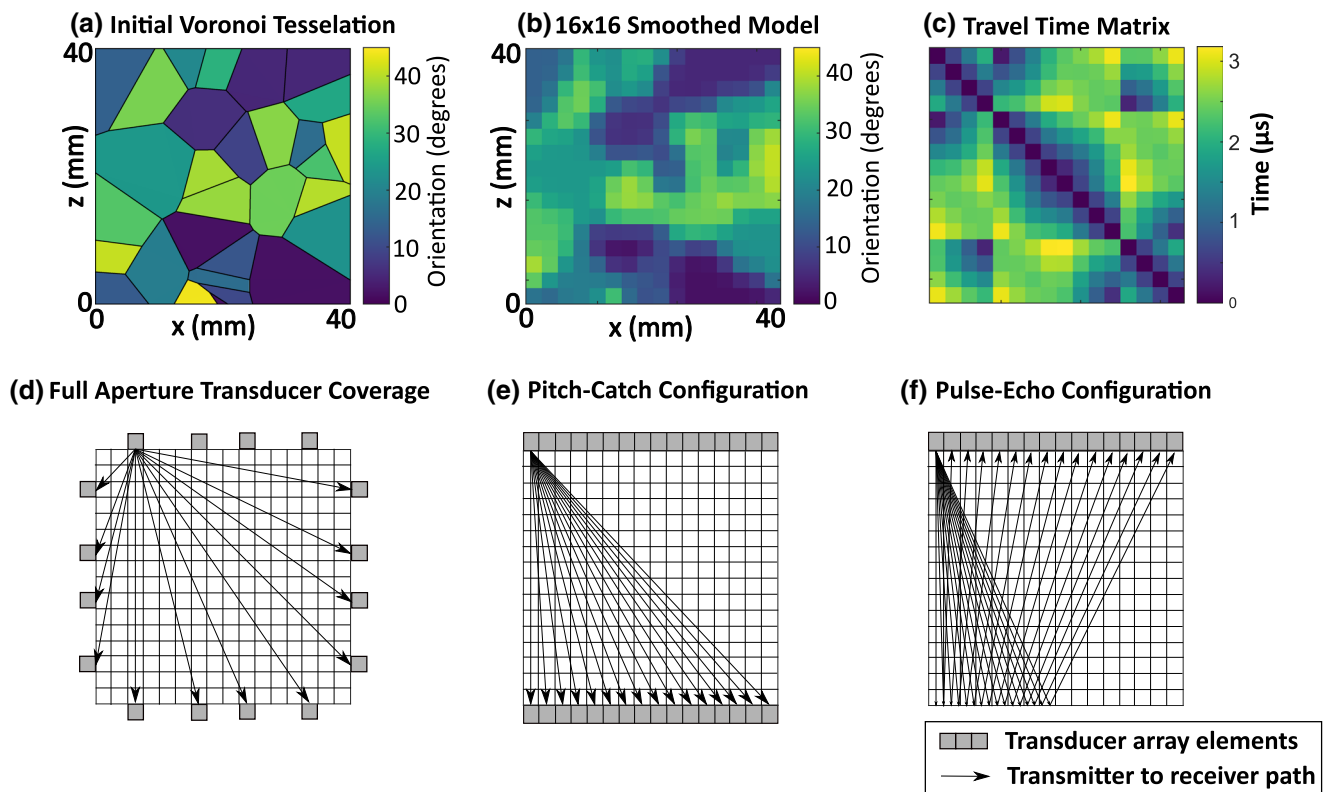Transducer array elements

→ Transmitter to receiver path

Fig. 1 Illustration of the procedure for generating training data: **a** Randomly generated Voronoi tessellation with 30 seeds and random grain orientations ranging between $0°$ and $45°$. (Other angles are included due to symmetry of the slowness curve.) **b** Material map $\mathbf{m}_{16}$ generated by discretising the Voronoi image **a** on a 16×16 grid and then smoothing with a Gaussian kernel. **c** An example travel time matrix $T_{\mathbf{m}_{16}}$ populated by measurements of the time of flight between every pair of transducer elements. In this paper, we position transducer elements in three configurations: **d** full aperture coverage by 16 elements (4 on each face), **e** two-sided aperture coverage with 16 transmitting elements at the top of the model with the pitch-catch time of flight being measured at 16 receiving elements at the bottom of the model and **f** one-sided aperture coverage, where 16 transducers are positions along the top face and the pulse-echo arrival time of the wave reflecting from the bottom face is measured

more computationally expensive than the AMSFMM. As a large number of data-model pairs are required to train a deep neural network, the more efficient AMSFMM method is used to generate travel time matrices $\mathbf{T_m}^{FMM}$ for the training data. The more physically realistic FEA generated data are then used to generate data to test the trained networks' performance (see FEA set-up in "Appendix"). Alternatively, a finite difference approach could be used for forward modelling wave propagation, which can provide similar levels of accuracy and computational cost to FEA; however, finite difference methods can be challenging to extend to irregular component geometries. A total of 7500 models are generated, and the corresponding travel time matrices $\mathbf{T_m}^{FMM}$ are computed using AMSFMM for the training data set, where additional 4 models are used with the FEA for testing purposes.

2.2 Deep neural network for orientation mapping

Deep neural networks (DNNs) are mathematical mappings that emulate the relationship between two parameter spaces [20]. Here, we seek a map between the grain orientation models $\mathbf{m}_{16}$ and the corresponding time of flight data $\mathbf{T_m}$ (that is $DNN(T_{16}^{FMM}) = \mathbf{m}_{16}^{pred}$, where the pred superscript denotes the DNN prediction). For each of the transducer configurations $\mathbf{s}$, a different number of travel times are used as input to the neural network. For a full aperture configuration (Fig. 1d), we have n source–receivers per side of our rectangular domain, and so there are $6n^2$ unique travel times (accounting for source–receiver reciprocity and excluding those between elements which lie on the same side). When $n = 4$, a set of 96 travel times is taken from each ToF matrix $\mathbf{T_m}$. For the pitch-catch configuration (Fig. 1e), all source–receiver paths are unique; therefore, the full ToF matrix is used, and with $n = 16$, there are 256 inputs to the neural network. Finally, for the pulse-echo

configuration (Fig. 1f), when n is even, there are $n^2/2 + n/2$ unique travel times (accounting for source–receiver reciprocity), so when $n = 16$, a total of 136 travel times are selected for the ToF matrix. For network training, both the input travel times and the output orientations are scaled to have zero mean and unit variance.

We configure three DNNs (corresponding to three transducer configurations), each with five fully connected layers (illustrated in Fig. 2), using sigmoid activation functions. The final output layer contains a single node corresponding to the orientation of a single pixel in the imaging domain. Therefore, following the approach of [20], a separate network is trained for each pixel, so for a 16×16 resolution image a total of 256 networks are trained. Alternatively, a single network with an output layer consisting of the same number nodes as pixels can be trained; however, the size of network and trainable parameters will be higher, and therefore, there would be a slower training process. The approach of training a separate network per pixel also allows individual network architectures to be modified for different pixels (though this is not considered in this study). The networks are trained using the Adam optimisation algorithm [38]. While a wide range of more sophisticated algorithms could be implemented to provide greater training accuracy (e.g. [28, 50]), this algorithm is implemented as it provides fast convergence and simple implementation. A description of network hyper-parameters is provided in "Appendix 2". These hyper-parameters are selected using a stochastic optimisation library [7] for each network architecture corresponding to different transducer configurations. We use a mean-squared-error (MSE) loss function, given by:

$$\text{MSE} = \frac{\sum_{i=1}^{N} \left(\mathbf{m}_{16}^{\text{true}}(i) - \mathbf{m}_{16}^{\text{pred}}(i) \right)^2}{N}, \qquad (2)$$

where $\mathbf{m}_{16}^{\text{true}}$ and $\mathbf{m}_{16}^{\text{pred}}$ are the true and predicted grain orientation models, i denotes the pixel index, and N is the total number of pixels (for models $\mathbf{m}_{16}$, $N = 256$). The choice of loss function controls the performance of the trained network. MSE penalises large prediction errors, whereas mean absolute error (MAE) is less sensitive to outliers. Alternatively, the structural similarity index measure (SSIM, [61]) could be used to penalise perceived changes in structural information, or the Wasserstein distance [18] could be used to emphasise the correct location of anomalous regions in the reconstructed images.

A validation data set is created using 20% of the training data. To avoid over-fitting the network to the training data, the cost function is periodically evaluated over the validation data set, and we implement an early stopping algorithm so that training stops once the validation loss stops decreasing (with a patience of 10 iterations). The time to train 256 separate networks sequentially using *Google Colab* [10] free graphics processing units (GPUs) is approximately 40 min, although training could be parallelised to reduce this time if required. Once trained, the compute time of $\mathbf{m}_{16}^{\text{pred}}$ is approximately 0.15 s per model inversion.

2.3 Generative adversarial networks for super-resolution

Conditional GANs learn a mapping between two images [30] and so can be used for post-processing of the DNN tomography output ($\mathbf{m}_{16}^{\text{pred}}$) to increase resolution and
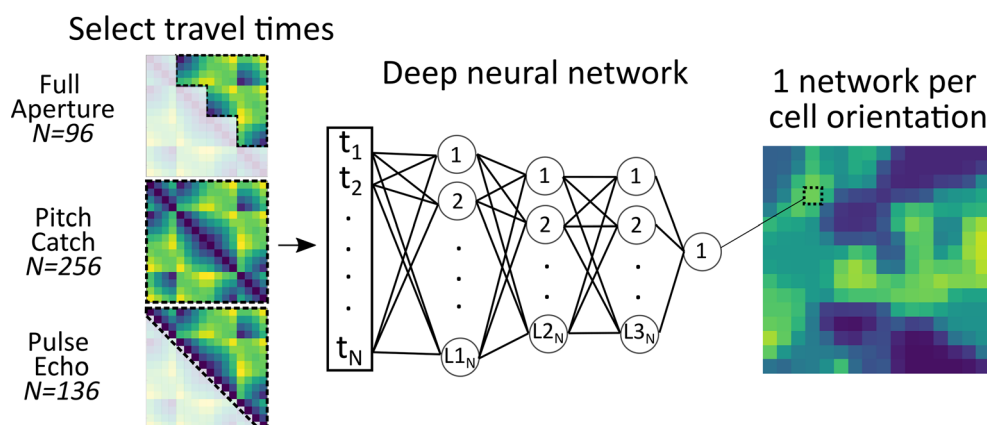

Fig. 2 Schematic of the tomography algorithm using deep neural networks (DNNs). First, N travel times are selected from the travel time matrix (where $N = 96$, $N = 256$ and $N = 136$ for the full aperture, pitch-catch and pulse-echo configurations, respectively). Travel times are then used as input into the DNN. The DNN consists of 3 hidden layers ($L1$, $L2$ and $L3$) and a final output layer. The nodes are illustrated as circles, and the number of nodes in each layer is denoted in the bottom circles ($L1_N$, $L2_N$ and $L3_N$). Each output corresponds to the crystal orientation of a single pixel in the material map $\mathbf{m}_{16}$, so 256 separate networks are trained in order to predict all pixel orientations

accuracy. The GAN architecture, as illustrated in Fig. 3a, consists of two separate trainable networks: a generator (GAN_G) and a discriminator (GAN_D).

Training a GAN for post-processing the output of the DNN tomography method ($\mathbf{m}_{16}^{\text{pred}}$) to achieve an increase in image resolution (super-resolution) requires an additional training data set, where travel time data are generated using higher-resolution models (64×64). The GAN framework assumes some prior knowledge of the structure of the material which is incorporated into the GAN training data.

For example, in layered structures such as carbon fibre-reinforced polymers (CFRPs), the training data should include models with locally anisotropic layers, or alternatively models exhibiting crystalline grain structures should be used to train GAN's for cases such as welds, and knowledge on the average grain size could feed into the complexity of the models included in the training data. We use three separate training data sets of increasing complexity. The first high-resolution model $\mathbf{m}_{64}^{\text{true}}$ consists of up to 5 horizontal layers where the orientation and thickness of each layer are randomly assigned (Fig. 3b). The second
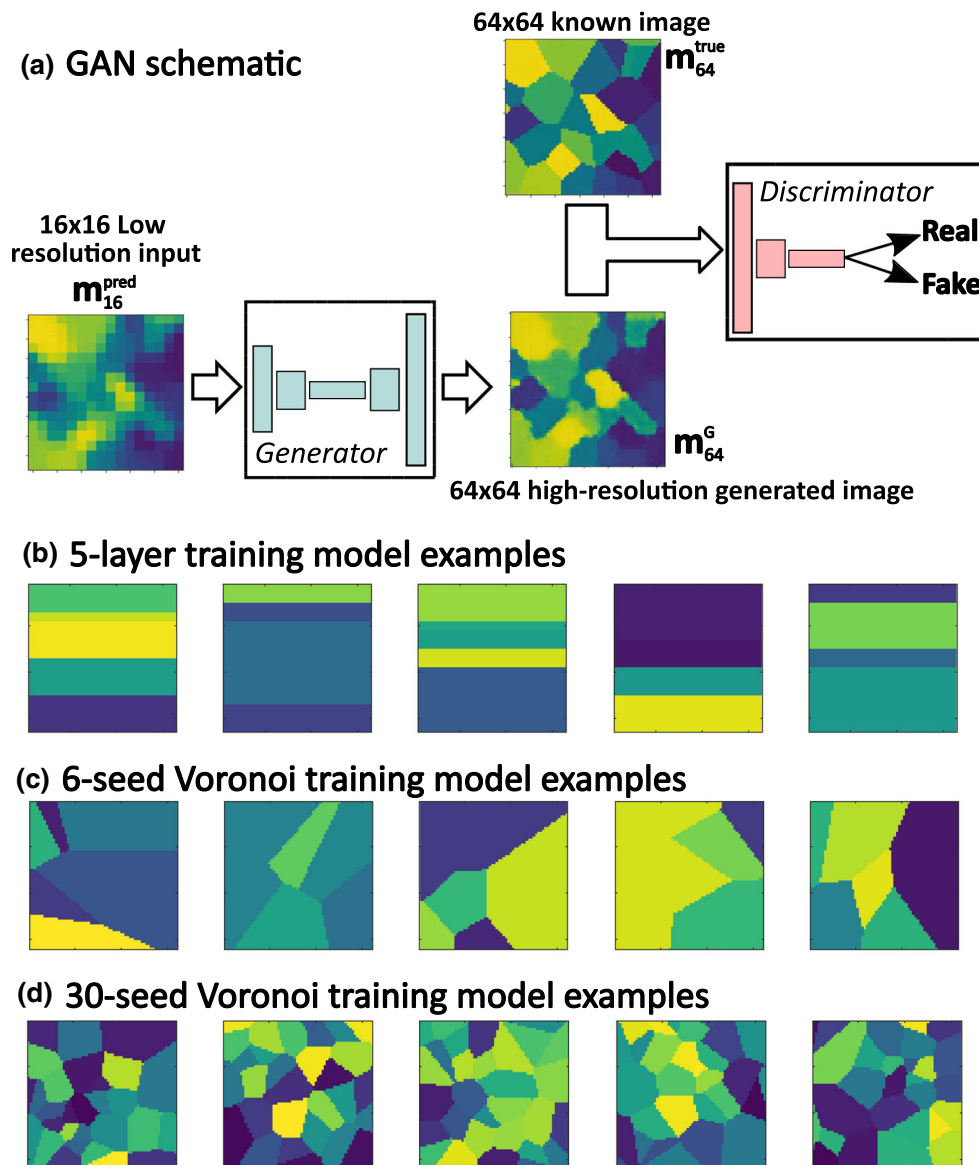

Fig. 3 **a** Schematic of the generative adversarial network (GAN) post-processing algorithm for achieving super-resolution material maps. The 16×16 output of the DNN tomography algorithm $\mathbf{m}_{16}$ is input to the generator network, which outputs a 64×64 image $\mathbf{m}_{64}^T$. The known 64×64 image $\mathbf{m}_{64}^T$, which was used to generate the DNN input data, as well as the output of the generator $\mathbf{m}_{64}^G$ is input into the discriminator, which outputs a prediction of which image is generated and which belongs to the training data set. Three different model data sets are used for GAN training: **b** a layer model with up to 5 layers where positions of the interfaces are random, **c** a random Voronoi tessellation with 6 seeds and **d** a random Voronoi tessellation with 30 seeds

and third are generated by discretising random Voronoi tessellations with 6 and 30 seed locations into a 64×64 grid, as shown in Fig. 3c and d, respectively. The travel time matrices $\mathbf{T_m}_{64}^{FMM}$ are calculated using the AMSFMM algorithm for 2000 models for each of the three data sets, which are input into the DNN tomography algorithm described in the previous section, which outputs a 16×16 predicted model $\mathbf{m}_{16}^{pred}$. The generator is configured to take the low-resolution $\mathbf{m}_{16}^{pred}$ image as input and to output a high-resolution 64×64 image $\mathbf{m}_{64}^{G}$ (i.e. $\mathrm{GAN}_G(\mathbf{m}_{16}^{pred}) = \mathbf{m}_{64}^{G}$). Here, the generator is a modified U-net [52] based on fully convolutional layers (see "Appendix" for network architecture). The discriminator takes the output of the generator $\mathbf{m}_{64}^{G}$, as well as the known 64×64 high-resolution image ($\mathbf{m}_{64}^{true}$) that was used to generate the ToF data, and predicts which image is generated (fake) and which is part of the training data (real). The accuracy of the discriminator prediction can then be established. These competing networks are then trained against each other; in each iteration of training, the accuracy of the discriminator is fed into the loss function of the generator network. The generator seeks to create images $\mathbf{m}_{64}^{G}$ that decrease the discriminator accuracy meaning that $\mathbf{m}_{64}^{G}$ cannot be discriminated from the reference training data $\mathbf{m}_{64}^{true}$. Following the training process, the generator can be used to map from 16×16 images to 64×64 resolution images.

3 Results

3.1 DNN results

Following the training of the fully connected DNN, we predict material maps $\mathbf{m}_{16}^{pred}$ using the three transducer array configurations shown in Fig. 1 following

$$\mathbf{m}_{16}^{pred} = \mathrm{DNN}\left(\mathbf{T_m}^{FMM}\right), \tag{3}$$

where $\mathbf{T_m}^{FMM}$ is test data which has not been used in the network training process. The test data are generated following the same protocol as for the training data, using smoothed Voronoi models $\mathbf{m}_{16}$ and the AMSFMM algorithm to generate a total of 200 test models and data. Comparisons of the true models $\mathbf{m}_{16}^{true}$ with the predicted models $\mathbf{m}_{16}^{pred}$ using the DNN and with full aperture, pitch-catch and pulse-echo transducer array configurations are shown in Fig. 4. We use two metrics for comparing predicted models with the true models: the mean absolute error (MAE), which is a scalar value (MAE ≥ 0, where MAE $= 0$ describes a perfect prediction), and the structural similarity index measure (SSIM) [61] ($-1 \leq \mathrm{SSIM} \leq 1$,

where SSIM $= 1$ describes a perfect prediction). The SSIM incorporates the similarity of three independent parameters: image luminescence, contrast and structure (see "Appendix"). These values are calculated with orientations that are scaled to have zero mean and unit variance. Note that *lower* values of MAE indicate higher similarity between the true and predicted models, whereas *higher* values of SSIM indicate higher image similarity.

In all cases, the predicted material property maps resemble the true orientation maps, predicting the magnitude and location of areas with similar orientations. The DNN predictions with a full aperture experimental configuration (Fig. 4b) perform the best (lower MAEs and higher SSIMs), and predictions made using the pulse-echo configuration perform the worst (higher MAEs and lower SSIMs). The histograms of MAE and SSIM values for the 200 test models are shown in Fig. 5a and b. The distributions of the pixel mean absolute error (averaged for each pixel across the 200 models) are shown for each transducer array configuration in Fig. 5c–e, showing that reconstruction accuracy generally decreases (increasing pixel MAE) in the central region of the domain and with distance from the transmitting element transducer array.

So far, the same mathematical model has been used for both the training data and the test data (a so-called inverse crime [62]), and this is not a sufficient challenge of the methodology [35]. We therefore now use a different mathematical model to test the trained DNN. One further additional challenge is to generate material maps using a different method from that used in the training data, so not originating from Voronoi diagrams. The material maps in Fig. 6a show a range of structures including a homogeneous model, a checkerboard structure, a layered structure and a single circular anomaly, all of which are significantly dissimilar from the textures and structures found within training data. The FEA method is used to generate ToF data $\mathbf{T_m}^{FEA}$ using a full aperture transducer array configuration, which is then input into the DNN to predict the grain orientation map $\mathbf{m}_{16}^{pred}$

$$\mathbf{m}_{16}^{pred} = \mathrm{DNN}(\mathbf{T_m}^{FEA}). \tag{4}$$

The predicted material maps $\mathbf{m}_{16}^{pred}$ shown in Fig. 6b and c show similar results using $\mathbf{T_m}^{FEA}$ and $\mathbf{T_m}^{FEA}$ time of flight data. In the cases of the homogeneous model and the single circular anomaly, the results using $\mathbf{T_m}^{FEA}$ are slightly improved (lower MAE). The similarity of results between the two data types indicates that the DNN is robust to changes in different data simulation methods and to the noise in the FEA data set associated with the identification of travel times. The presence of this additional noise does not appear to have a significant effect on the changes in measured travel time due to anisotropy, and therefore, the
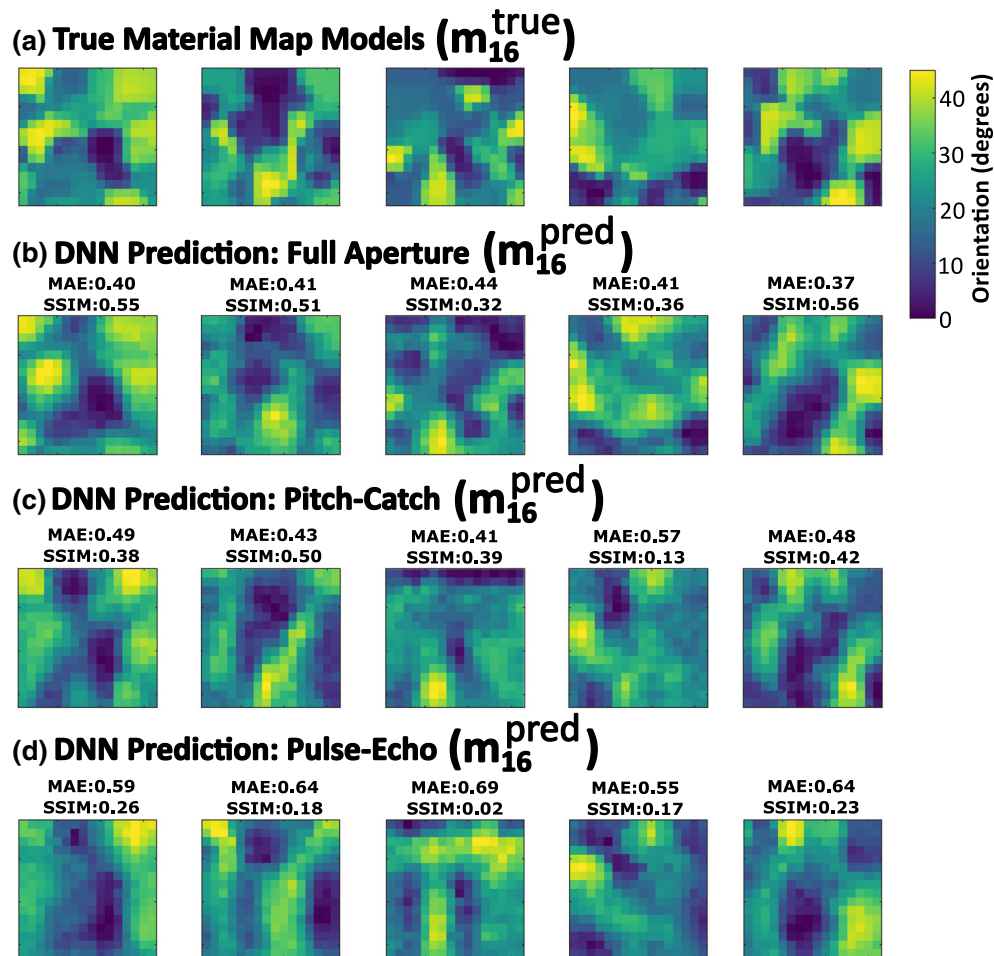
(a) True Material Map Models $(\mathbf{m}_{16}^{true})$

(b) DNN Prediction: Full Aperture $(\mathbf{m}_{16}^{pred})$

| MAE:0.40 | MAE:0.41 | MAE:0.44 | MAE:0.41 | MAE:0.37 |
| SSIM:0.55 | SSIM:0.51 | SSIM:0.32 | SSIM:0.36 | SSIM:0.56 |

(c) DNN Prediction: Pitch-Catch $(\mathbf{m}_{16}^{pred})$

| MAE:0.49 | MAE:0.43 | MAE:0.41 | MAE:0.57 | MAE:0.48 |
| SSIM:0.38 | SSIM:0.50 | SSIM:0.39 | SSIM:0.13 | SSIM:0.42 |

(d) DNN Prediction: Pulse-Echo $(\mathbf{m}_{16}^{pred})$

| MAE:0.59 | MAE:0.64 | MAE:0.69 | MAE:0.55 | MAE:0.64 |
| SSIM:0.26 | SSIM:0.18 | SSIM:0.02 | SSIM:0.17 | SSIM:0.23 |

Fig. 4 **a** True material orientation maps $\mathbf{m}_{16}^{true}$ from the test data set and corresponding predicted models $\mathbf{m}_{16}^{pred}$ using the DNN tomography algorithm for **b** full aperture, **c** pitch-catch and **d** pulse-echo transducer array configurations. The associated mean absolute errors (MAEs) and the structural similarity index metrics (SSIMs) between the predicted and true maps are labelled above each reconstructed material map

inversion remains accurate. The accuracy of the predicted models is lower where the material maps exhibit different textures to those used in the training data; compare the MAE and SSIM values in Fig. 6c with those in Fig. 4b. The higher accuracy of the results in Fig. 4b highlights that the texture of the target application material for the DNN tomography algorithm should be included as far as possible in the training data set.

3.2 GAN results

Three GANs are trained using the layered, 6-seed Voronoi and 30-seed Voronoi models $\mathbf{m}_{64}^{true}$, and 200 additional models per GAN are used for testing, of which 5 are shown in Figs. 7a, 8a and 9a, respectively. The AMSFMM method is used to compute travel time data ($\mathbf{T_m}^{FMM}$) using a full aperture transducer array configuration, which are input into the trained DNN (as used for the generation of

DNN predictions in Fig. 4b). The DNN predicted outputs $\mathbf{m}_{16}^{pred}$ are shown in Figs. 7b, 8b and 9b and the GAN outputs $\mathbf{m}_{64}^{G}$ in Figs. 7c, 8c and 9c for the layered, 6-seed Voronoi and 30-seed Voronoi models, respectively. In order for image comparison with MAE and SSIM, the 16×16 resolution DNN outputs are upscaled to 64×64 resolution using nearest neighbour interpolation. Histograms of the changes in MAE ($\Delta MAE = MAE_{GAN} - MAE_{DNN}$) and SSIM ($\Delta SSIM = SSIM_{GAN} - SSIM_{DNN}$) when using a GAN to post-process the DNN tomography outputs are shown in Fig. 10.

For the 5 layer models (Fig. 7), the GAN predictions are significantly more accurate compared to DNN predictions, offering large improvements in MAE (decrease up to $\Delta MAE = -0.85$) and SSIM (increase up to $\Delta SSIM = 0.5$). The GAN successfully learns to generate horizontal (layered) structures, so very little horizontal variation exists in the GAN predictions. The reconstructed grain orientation
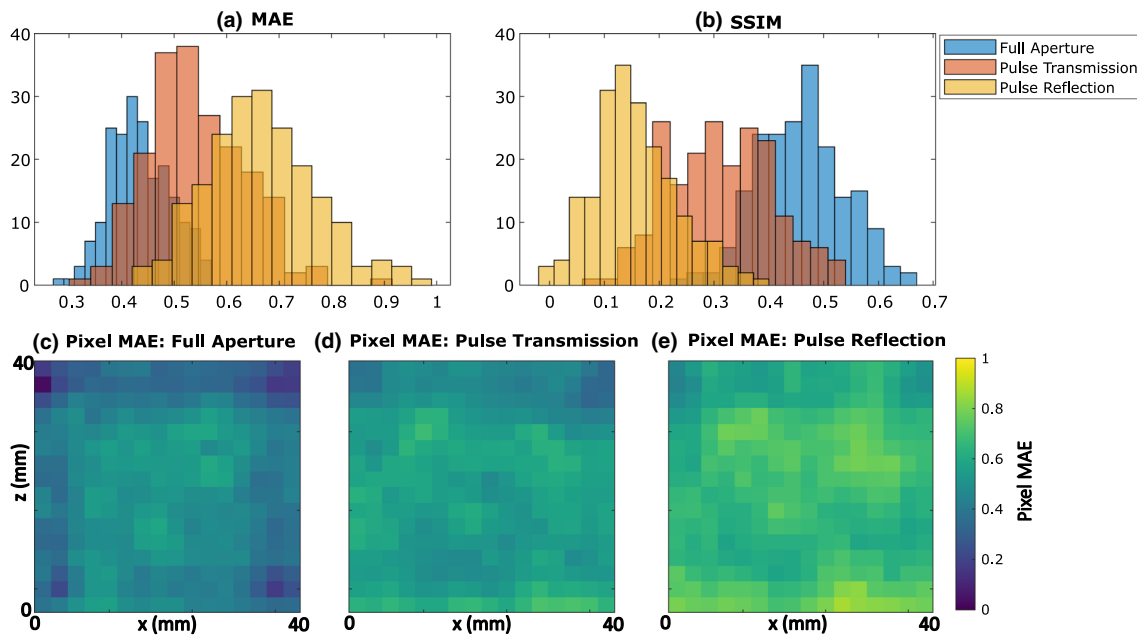
Fig. 5 Histogram of **a** mean absolute error (MAE) and **b** structural similarity index metric (SSIM) for DNN prediction on 200 test models using different transducer array configurations. Lower values of MAE and higher values of SSIM suggest a good tomographic reconstruction. The lower panels show the mean absolute error (MAE) for each pixel averaged across the 200 test models for **c** full aperture, **d** pitch-catch and **e** pulse-echo transducer array configurations
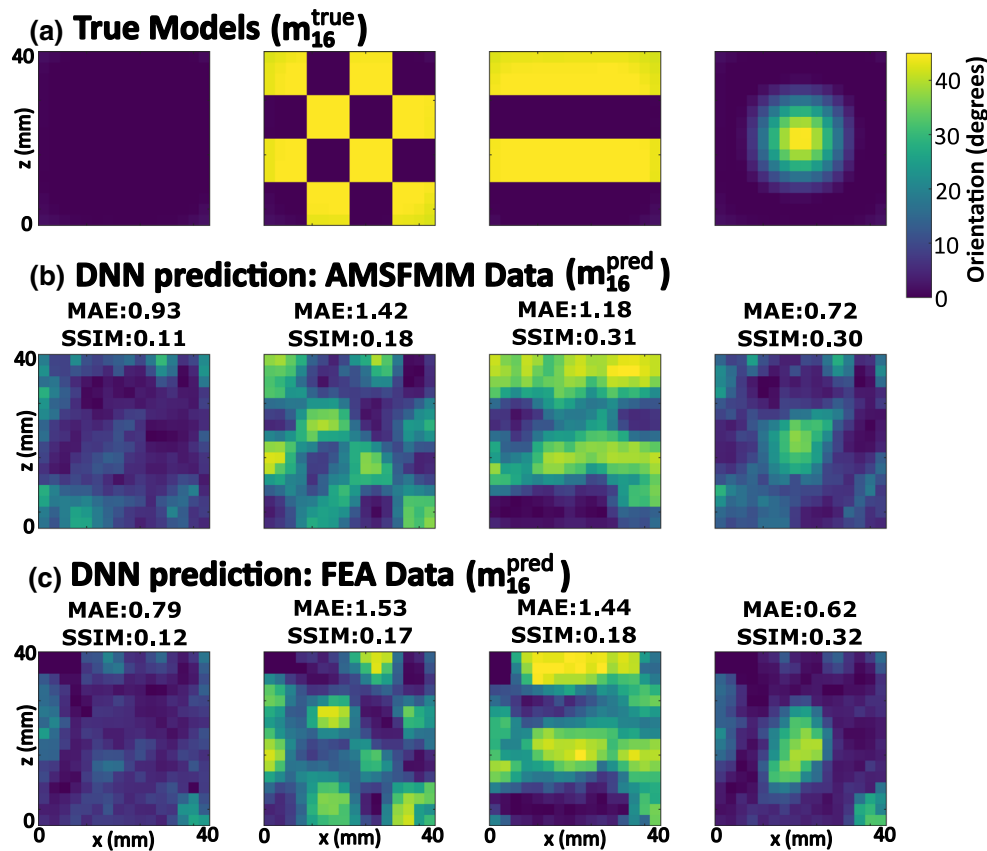

Fig. 6 a True material maps ($\mathbf{m}_{16}^{\text{true}}$). The predicted models using the AMSFMM trained DNN tomography algorithm with **b** $\mathbf{m}_{16}^{\text{pred}} = \text{DNN}(\mathbf{T_m}^{\text{FMM}})$ and **c** $\mathbf{m}_{16}^{\text{pred}} = \text{DNN}(\mathbf{T_m}^{\text{FEA}})$
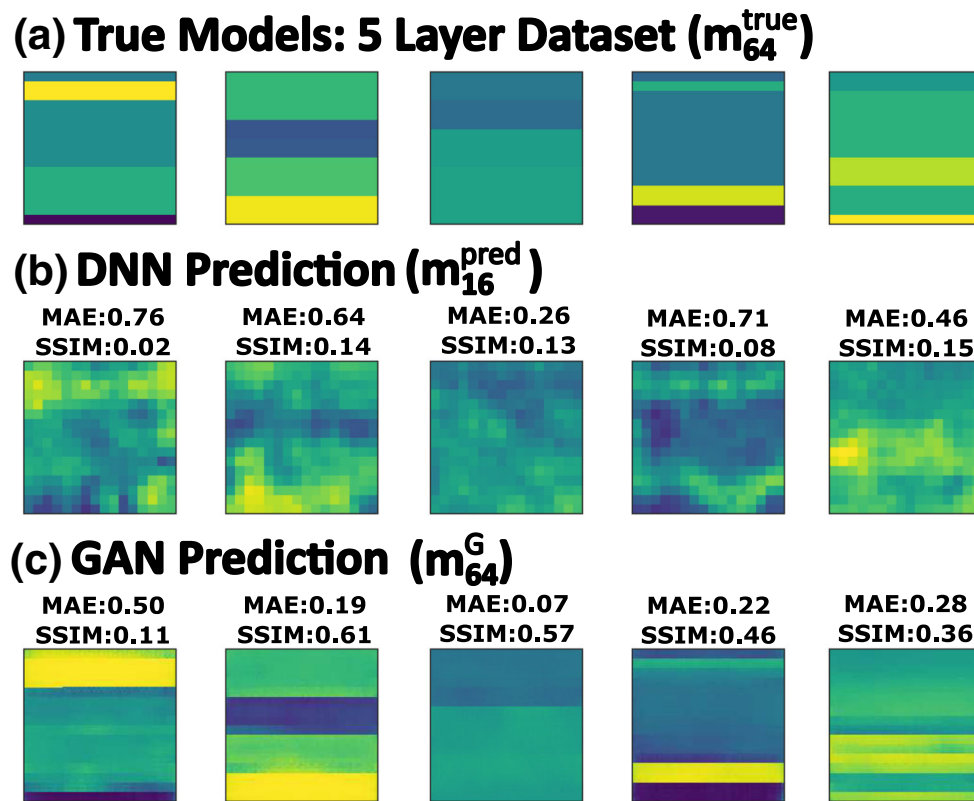
(a) True Models: 5 Layer Dataset (m_{64}^{true})

(b) DNN Prediction (m_{16}^{pred})

| MAE:0.76 | MAE:0.64 | MAE:0.26 | MAE:0.71 | MAE:0.46 |
| SSIM:0.02 | SSIM:0.14 | SSIM:0.13 | SSIM:0.08 | SSIM:0.15 |

(c) GAN Prediction (m_{64}^{G})

| MAE:0.50 | MAE:0.19 | MAE:0.07 | MAE:0.22 | MAE:0.28 |
| SSIM:0.11 | SSIM:0.61 | SSIM:0.57 | SSIM:0.46 | SSIM:0.36 |

Fig. 7 **a** True high-resolution (64×64) grain orientation maps m_{64}^{true} consisting of 5 horizontal layers, **b** 16×16 resolution DNN tomography output (m_{16}^{pred}) and **c** 64×64 GAN output m_{64}^{G}. For row **b**, the MAE and SSIM are calculated on an upscaled image to 64×64 resolution using nearest neighbour interpolation. Note the significant improvements in MAE and SSIM using the GAN methodology and the clear improvements in reconstructing a layered structure

maps from the GAN exhibit discontinuous grain boundaries and piecewise constant orientations for each layer, compared to the smooth spatially varying DNN tomography outputs (Fig. 7b). The GAN also performs well for the 6-seed Voronoi tessellation models (Fig. 8), where reconstructed grain orientation maps from the GAN exhibit discontinuous, piecewise constant orientations for each grain. The GAN improves MAE and SSIM in all cases; however, there is slight blurring across some grain boundaries. The GAN results for the 30-seed Voronoi tessellation models (Fig. 9) exhibit stronger blurring across grain boundaries. While the GAN prediction is texturally more similar to the true models (piecewise constant and discontinuous regions), the distributions of ΔMAE and ΔSSIM in Fig. 10 show the GAN offers only marginal improvements in reconstruction accuracy, and in some cases the accuracy decreases when using the GAN (ΔMAE > 0 and ΔSSIM < 0). The difference between the 6-seed and 30-seed Voronoi models is in the model complexity due to smaller individual grains in the 30-seed models. In these models, multiple grains can fit into a single pixel of a low-resolution DNN tomography image, resulting in a loss of spatial information that the GAN

cannot fully recover. These results show that a GAN can be used for post-processing tomography results to improve reconstruction accuracy and image resolution, particularly when prior information regarding the spatial distribution of the material map is known (e.g. if the sample is known to be layered, or similarly well-structured) and the spatial distribution is simple.

4 Discussion

The framework presented includes several stages: (1) the generation of training data using the AMSFMM method, (2) training of the DNN, and (3) training of the GAN. However, each of these stages only need be performed once. Thereafter, the DNN and GAN can be used in effectively real time (<1 s). Here, the time for generating 7500 ToF matrices T_m^{FMM} was approximately 1 h, for training the DNN was approximately 40 min (until convergence), and for training the GAN was approximately 8 h (using *Google Colab* GPUs [10]). There are several alternative algorithms for solving the tomographic inverse problems, which would require less computing time than
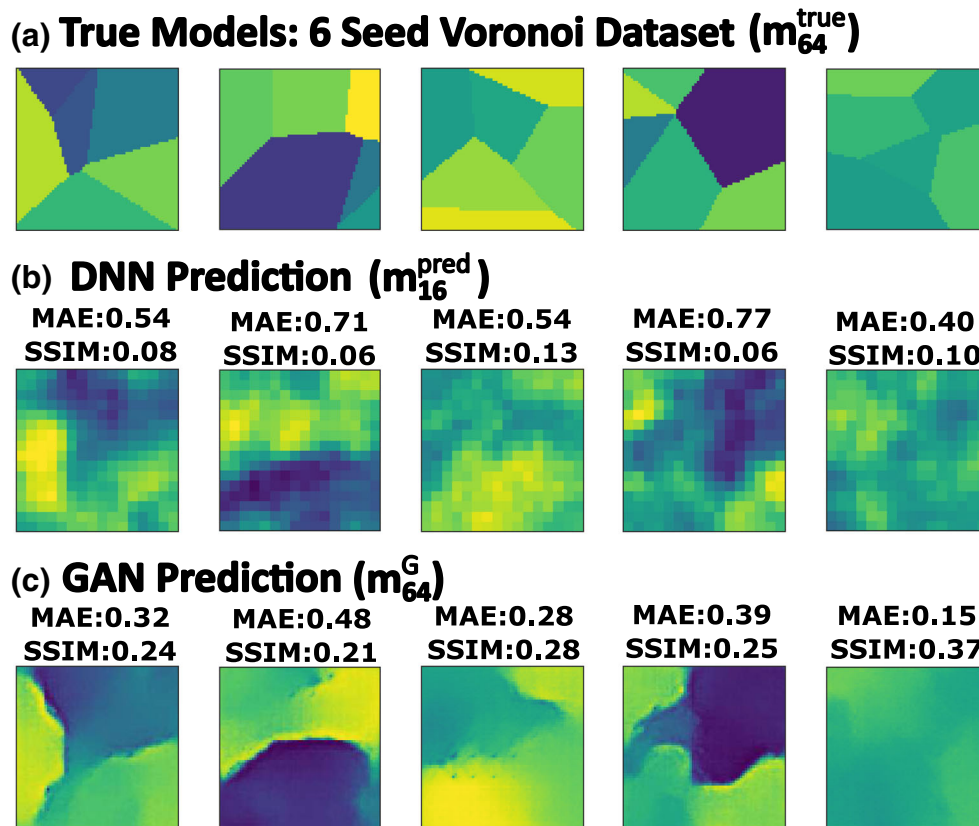
(a) True Models: 6 Seed Voronoi Dataset ($\mathbf{m}_{64}^{true}$)

(b) DNN Prediction ($\mathbf{m}_{16}^{pred}$)

MAE:0.54 SSIM:0.08 MAE:0.71 SSIM:0.06 MAE:0.54 SSIM:0.13 MAE:0.77 SSIM:0.06 MAE:0.40 SSIM:0.10

(c) GAN Prediction ($\mathbf{m}_{64}^{G}$)

MAE:0.32 SSIM:0.24 MAE:0.48 SSIM:0.21 MAE:0.28 SSIM:0.28 MAE:0.39 SSIM:0.25 MAE:0.15 SSIM:0.37

Fig. 8 **a** True high-resolution (64×64) grain orientation maps $\mathbf{m}_{64}^{true}$ consisting of a Voronoi tessellation with 6 seeds, **b** 16×16 resolution DNN tomography output ($\mathbf{m}_{16}^{pred}$) and **c** 64×64 GAN output $\mathbf{m}_{64}^{G}$. For row **b**, the MAE and SSIM are calculated on an upscaled image to 64×64 resolution using nearest neighbour interpolation. Note the significant improvements in MAE and SSIM using the GAN methodology and the clear improvements in reconstructing piecewise constant structures

for the deep learning training process (e.g. genetic algorithms [24], conjugate gradient least squares algorithms [46] and the simultaneous iterative reconstruction technique [57]). These algorithms involve sampling possible models multiple times (e.g. a family of 20 models for 100 generations using a genetic algorithm as in [24]), and depending on the speed of the forward model technique, this process can range from minutes to hours of compute time. While this may be faster than the DNN training process, a repeated inversion (as is required for monitoring) would require the whole process to be restarted, whereas the DNN approach is real time once trained. It is clear that when repeated material map reconstructions are desired, as is the case for NDE monitoring purposes, the deep learning framework excels in its ability to provide real-time results. There is therefore also a strong potential to extend the capabilities of the current framework to include spatio-temporal modelling by incorporating long short-term memory (LSTM) networks into the network architecture [23].

The benefits of real-time inversions come at the expense of a few limitations that are yet to be overcome in the current work. Firstly, the DNN is trained with a constant transducer configuration and on a limited set of training data, so a trained DNN cannot be generally extended to changes in relative transducer locations or be used to reconstruct materials whose properties are not present in the training data. This is not a problem for many applications in NDE, as the transducer arrays are rigid and fixed, and the test sample geometries do not change through time. However, limited network flexibility may be problematic in cases where the configuration changes, such as in-process monitoring of additive manufacturing: during the building process, the shape of the sample changes therefore the distribution of transducer elements also changes. One solution is to train many DNNs for all the possible transducer configurations throughout the building process; however, this would require a significantly expensive training process. Another solution, proposed in [20], is to train more flexible networks that account for missing data by augmenting the training data set with additional input samples taken from additional transducer locations. Travel times in the ToF matrix can be set to zero to indicate that a transducer is not used for a particular transducer
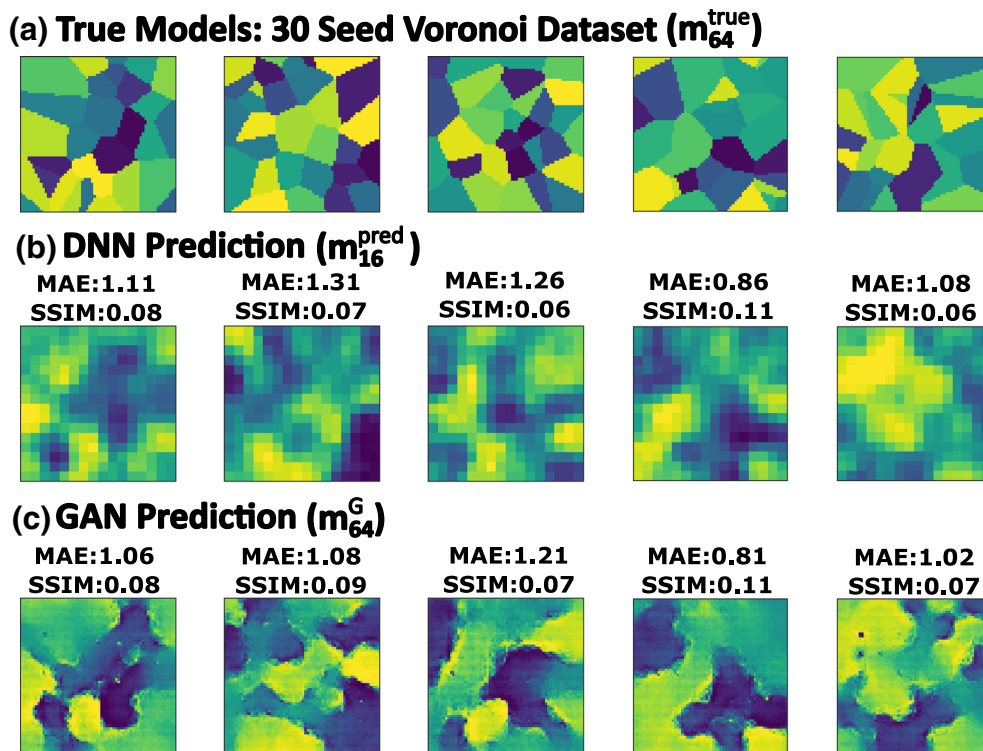
(a) True Models: 30 Seed Voronoi Dataset (m_{64}^{true})

(b) DNN Prediction (m_{16}^{pred})

| MAE:1.11 | MAE:1.31 | MAE:1.26 | MAE:0.86 | MAE:1.08 |
| SSIM:0.08 | SSIM:0.07 | SSIM:0.06 | SSIM:0.11 | SSIM:0.06 |

(c) GAN Prediction (m_{64}^{G})

| MAE:1.06 | MAE:1.08 | MAE:1.21 | MAE:0.81 | MAE:1.02 |
| SSIM:0.08 | SSIM:0.09 | SSIM:0.07 | SSIM:0.11 | SSIM:0.07 |

Fig. 9 a True high-resolution (64 × 64) grain orientation maps m_{64}^{true} consisting of a Voronoi tessellation with 30 seeds, **b** 16 × 16 resolution DNN tomography output (m_{16}^{pred}) and **c** 64 × 64 GAN output m_{64}^{G}. For row **b**, the MAE and SSIM are calculated on an upscaled image to 64 × 64 resolution using nearest neighbour interpolation. Note the marginal improvements in MAE and SSIM using the GAN methodology

configuration, and then, the trained network can invert using multiple configurations.

The GAN is also limited in its applicability. This is highlighted when a trained GAN is used to invert for textures that are dissimilar to those found in the training data. This can be seen in Fig. 11, where the GAN trained on the 30-seed Voronoi models is applied to the DNN prediction of the checkerboard, layered and circular inclusion material models as well as a 30-seed Voronoi model for reference. There are significant decreases in SSIM and increases in MAE when using the GAN on the models with dissimilar textures to the Voronoi models. This highlights the importance of the data used to train the network and suggests that the GAN should only be used if prior knowledge of the material is known and the expected textures are present in the training data. In the case of NDE, it is realistic for this prior information to be known (for example, an NDE operator will know whether the material of interest is a laminar composite or a welded steel). However, training a GAN with a much broader training data set, for example, including all of the layered, 6-seed and 30-seed Voronoi models in the same training data set, would allow for more general application of the GAN where less prior knowledge of the material is known. We leave this for future work.

Where real-time inversions are not required, more computationally expensive tomography algorithms can be implemented. Algorithms such as the reversible-jump Markov chain Monte Carlo [55] offer more information including an estimate of the uncertainty of the tomography results. A place for rapid deep learning-based tomography still exists within this framework as it can provide a fast, coarse initial model which can be used a starting point for more sophisticated algorithms. Additionally, a GAN can be used in post-processing any tomographic image. Often linearised image methods are often regularised and hence predict smoother structures that are expected to exist in the true medium, and therefore, a GAN can be trained to upscale resolution and sharpen these images. Even where the GAN provides marginal improvements to the DNN
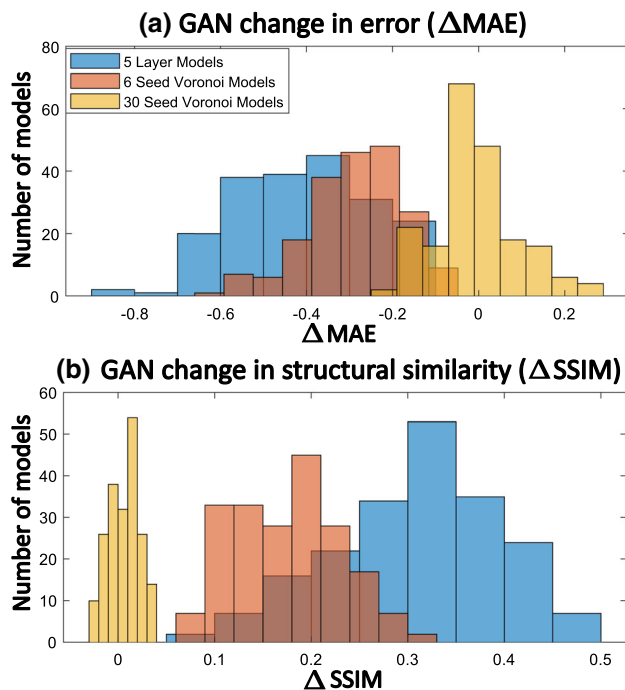
(a) GAN change in error (ΔMAE)

(b) GAN change in structural similarity (ΔSSIM)

Fig. 10 Histograms showing the change in **a** the mean absolute error (MAE) and **b** the structural similarity index measure (SSIM) when using a GAN to post-process 200 low-resolution DNN tomography outputs ($\Delta MAE = MAE_{GAN} - MAE_{DNN}$ and $\Delta SSIM = SSIM_{GAN} - SSIM_{DNN}$)

tomography results, the GAN output models exhibit discontinuous boundaries. It can be important that such boundaries are present in tomography algorithms where entire waveforms are modelled and matched to the recorded waveforms (that is, full waveform inversion [58]). A GAN might also be extended to take the full waveform as an input, though this would require expensive FEA modelling to generate the training data, so that all internal reflections are modelled.

The purpose of this study is to present a framework and its capabilities rather than a complete, optimised network. Therefore, there are several steps that can be taken to further improve the performance of the DNN and GAN networks [48]. The networks in this study use simple loss functions such as MSE and MAE. The Wasserstein (or Earth-Mover) distance emphasises accurate reconstruction of the location of spatial information as well as of the absolute parameter values. A Wasserstein-GAN approach [6] could yield better reconstruction performance compared to a MAE based GAN and would make for an interesting future study. Estimating the uncertainty of the predicted tomographic images, for example, through the

use of mixture density networks [20] or Bayesian neural networks [34], is an important future step to improve the current framework, as this would give an indication of cases where predictions are made on materials that are not represented in the training data. To reach the capability where the DNN and GAN tomography framework can be applied to experimentally acquired data, several assumptions need to be overcome. For example, our method assumes there is no background noise in the acquired data and no uncertainty in the transducer locations. As these assumptions may not be valid for experimentally acquired data, additional noise can be incorporated into the modelled travel time data representing the level of background noise or the uncertainty in the transducer locations. The framework should then be tested on acquired data from a controlled physical experiment, where a ground truth of material properties is available. Estimating the sensitivity of the resulting tomographic images to perturbations of the training hyper-parameters is also important step that should be taken to improve the current framework

5 Conclusion

We present a deep learning-based framework for the real-time tomographic reconstruction of spatially varying crystal orientations in locally anisotropic media using ultrasonic array time-of-flight data. We train a series of deep neural networks (DNNs) using 7500 models in a training data set, to accurately reconstruct orientation maps using full aperture, pitch-catch and pulse-echo transducer array configurations. We present the first application of generative adversarial networks (GANs) on ultrasonic tomographic data, where a series of GANs are trained with three sets of training data exhibiting increasing levels of complexity in the model textures. The GAN takes the low-resolution DNN output and refines the resolution by a factor of four. We show that prior information used to create the training data for both the DNN and the GAN is important factors in providing accurate estimations of the orientation maps. The proposed framework is currently limited in its application to a set of fixed transducer array configurations and a fixed component shape. This can be overcome by augmenting the training data to represent a wider range of configurations. Providing a wider range of the types of textures in the training data will enhance the applicability of the GAN. Using the methods presented unlocks a wide range of potential applications for ultrasonic monitoring, allowing for faster and more accurate
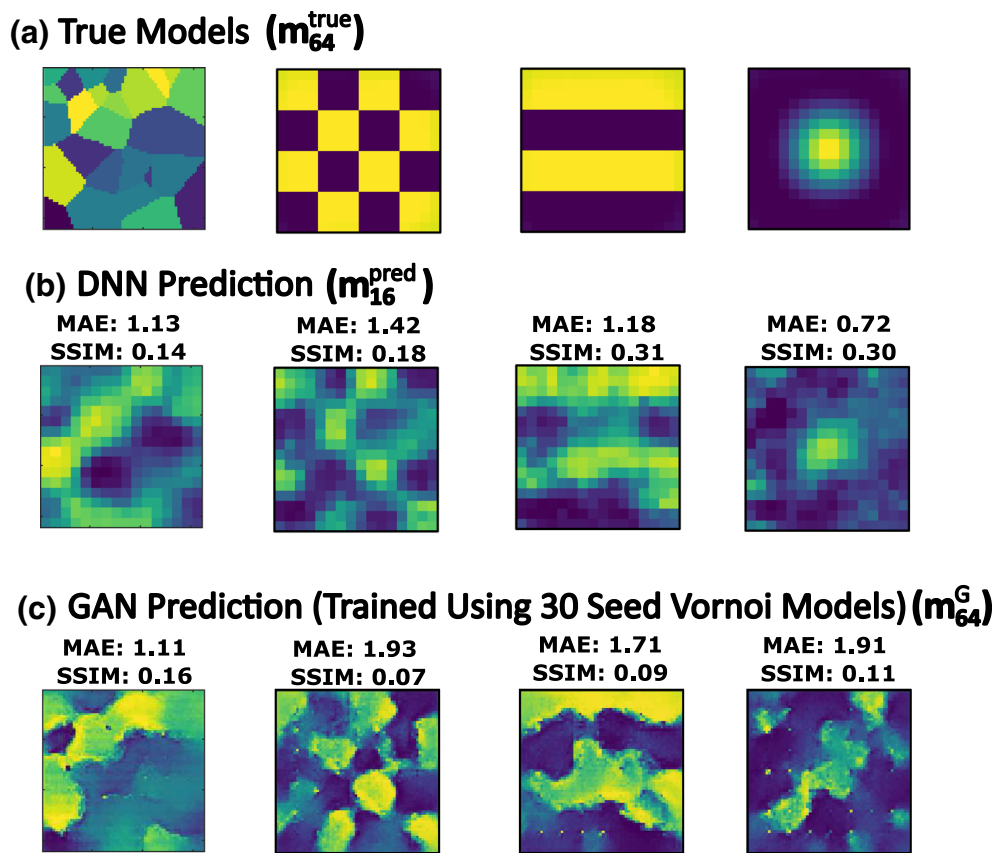
(a) True Models $(\mathbf{m}_{64}^{true})$

(b) DNN Prediction $(\mathbf{m}_{16}^{pred})$

| MAE: 1.13 | MAE: 1.42 | MAE: 1.18 | MAE: 0.72 |
| SSIM: 0.14 | SSIM: 0.18 | SSIM: 0.31 | SSIM: 0.30 |

(c) GAN Prediction (Trained Using 30 Seed Vornoi Models) $(\mathbf{m}_{64}^{G})$

| MAE: 1.11 | MAE: 1.93 | MAE: 1.71 | MAE: 1.91 |
| SSIM: 0.16 | SSIM: 0.07 | SSIM: 0.09 | SSIM: 0.11 |

Fig. 11 **a** True high-resolution (64 × 64) grain orientation maps $\mathbf{m}_{64}^{true}$, **b** 16 × 16 resolution DNN tomography output using AMSFMM generated data ($\mathbf{m}_{16}^{pred}$) and **c** 64 × 64 GAN output $\mathbf{m}_{64}^{G}$. For row **b**, the MAE and SSIM are calculated on an upscaled image to 64 × 64 resolution using nearest neighbour interpolation. It can be seen that the similar MAE and SSIM values result for the Voronoi diagram arises since this type of texture was used in the training data of the DNN and GAN. However, the GAN performs significantly worse in the cases where the material texture is not part of the training data in columns 2, 3 and 4

detection of flaws and in-process inspection during manufacturing.

Appendix

Finite element analysis

We implement a finite element simulation of elastic wave propagation in anisotropic media using OnScale [45]. We apply absorbing boundary conditions on all sides of the domain, so energy continues past boundaries with no reflections. We use Ricker wavelets with central frequencies of 1 MHz as the source-time function and apply pressure loads following the full aperture transducer array configuration as shown in Fig. 1d. The values for the finite element node spacing ($\Delta x, \Delta y$) are selected to ensure spatial stability conditions following $\Delta x, \Delta y = \frac{\lambda}{15}$, where λ is the shortest wavelength in the domain.

Following the simulation for each transmitting array element, the travel time to each receiving transducer is automatically picked by selecting the time for arriving energy to increase above a threshold. This threshold is taken to be 2% of the peak displacement in the recorded signal.

Network architectures

The deep neural networks (DNNs) are trained using 5 layers, where each node receives an input from every node in the previous layer and a sigmoidal activation function. The number of nodes in each layer is shown in Table 1 and other DNN hyperparameters are shown in Table 2.

The GAN generator is a modified U-Net based on [30] consisting of an encoder–decoder chain. Each block in the encoder is a convolution-batch normalisation-leaky rectified linear unit (ReLu) activation sequence. Each block in the decoder is a transposed convolution-batch normalisation-ReLu sequence with skip connections between mirrored layers in the encoder and decoder stacks [52] (as shown in Fig. 12a). All convolutional layers use a kernel size of 4. The generator loss is the discriminator sigmoid

Table 1 Network configurations showing the number of nodes for each layer including the three hidden layers (L1–L3) for the full aperture, pitch-catch and pulse-echo transducer array configurations

Array configuration	No. of inputs nodes	$L1_N$	$L2_N$	$L3_N$	No. of trainable parameters
Full Aperture	96	315	63	63	58,591
Pitch-Catch	256	354	55	55	113,911
Pulse-Echo	136	353	68	68	72,511

Table 2 Network hyper-parameters for the employed deep neural networks (DNN) and generative adversarial networks (GAN)

Hyper-parameter	DNN	GAN
Optimiser	Adam	Adam
Learning rate	4×10^{-3}	2×10^{-4}
Activation	Sigmoid	Rectified linear unit (ReLu) and Leaky ReLu
Loss function	MSE	MAE + Sigmoid cross-entropy (discriminator accuracy)
Minibatch size	80	5

Fig. 12 Network configurations for **a** the GAN generator U-net architecture and **b** the GAN discriminator PatchGAN architecture. All convolutional layers have a kernel size of 4
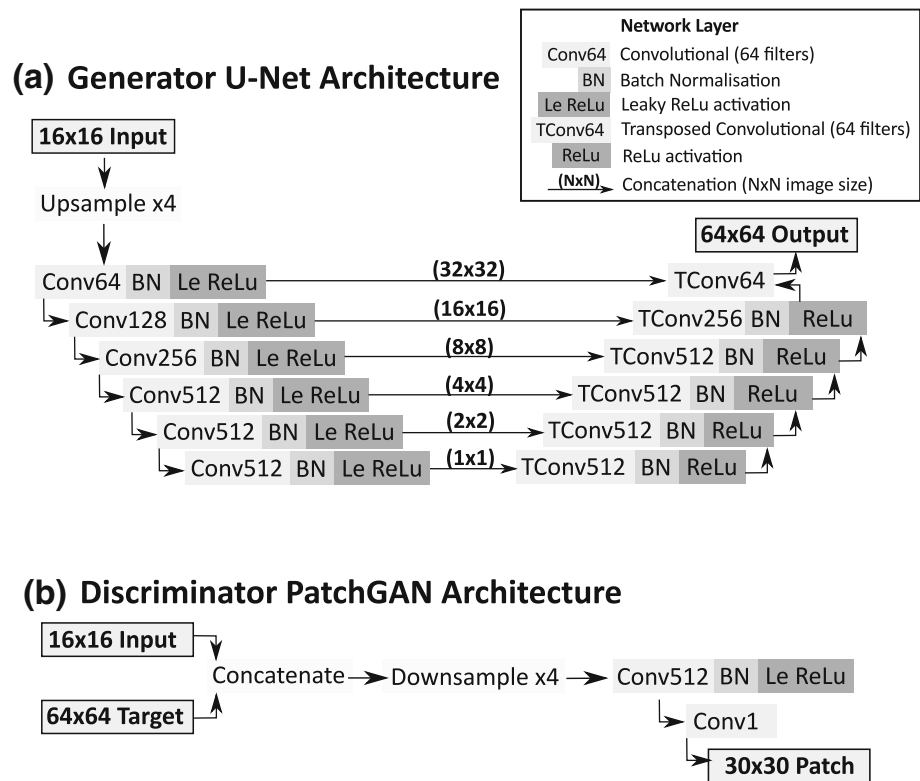
cross-entropy loss of the generated image with an array of ones combined with the mean absolute error between the generated and known target image (other GAN hyperparameters are provided in Table 2).

The GAN discriminator (Fig. 12b) follows a PatchGAN architecture [30], which divides the image into smaller 30×30 patches and the discriminator tries to classify each patch separately. This motivates the GAN to discriminate high-frequency structure. The discriminator receives the target and generated images as well as the low-resolution input. The discriminator loss is the sigmoid cross entropy loss with the real image and an array of ones, combined with the sigmoid cross entropy loss with generated image and an array of zeros.

Structural similarity index measure (SSIM)

We use the SSIM described by [61] for image comparison. The SSIM is defined as a weighted combination of comparisons between image luminance $l(X, Y)$, contrast $c(X, Y)$

and structure $s(X, Y)$, where X and Y describe an image window in known and estimated images of size $N \times N$. The SSIM is therefore

$$\text{SSIM}(X, Y) = [l(X, Y)]^{\alpha} \cdot [c(X, Y)]^{\beta} \cdot [s(X, Y)]^{\gamma} \qquad (5)$$

where α, β and γ are the weighting parameters. We use $\alpha = \beta = \gamma = 1$. Luminance, contrast and structure are calculated as

$$l(X, Y) = \frac{2\mu_X \mu_Y + C_1}{\mu_X^2 + \mu_Y^2 + C_1}, \qquad (6)$$

$$c(X, Y) = \frac{2\sigma_X \sigma_Y + C_2}{\sigma_X^2 + \sigma_Y^2 + C_2}, \qquad (7)$$

$$s(X, Y) = \frac{\sigma_{XY} + C_3}{\sigma_X \sigma_Y + C_3} \qquad (8)$$

where μ and σ are the mean and variance of the windows X or Y and σ_{XY} is the covariance of X and Y. This is computed over a sliding Gaussian window of 9×9.

Data and code availability The data and Python scripts required to reproduce these findings are available at: https://github.com/jonnyrsingh/DeepLearningAnisoTomo, which can be executed within Google Colaboratory. This requires no additional software or downloads for the user.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Aki K, Christoffersson A, Husebye ES (1977) Determination of the three-dimensional seismic structure of the lithosphere. J Geophys Res 82(2):277–296
2. Aki K, Lee W (1976) Determination of three-dimensional velocity anomalies under a seismic array using first P arrival times from local earthquakes: 1. A homogeneous initial model. J Geophys Res 81(23):4381–4399
3. Amiri N, Farrahi G, Kashyzadeh KR, Chizari M (2020) Applications of ultrasonic testing and machine learning methods to predict the static and fatigue behavior of spot-welded joints. J Manuf Process 52:26–34
4. Antholzer S, Haltmeier M, Schwab J (2019) Deep learning for photoacoustic tomography from sparse data. Inverse Probl Sci Eng 27(7):987–1005
5. Araya-Polo M, Jennings J, Adler A, Dahlke T (2018) Deep-learning tomography. Lead Edge 37(1):58–66
6. Arjovsky M, Chintala S, Bottou L (2017) Wasserstein generative adversarial networks. In: International conference on machine learning. PMLR, pp 214–223
7. Bergstra J, Komer B, Eliasmith C, Yamins D, Cox DD (2015) Hyperopt: a Python library for model selection and hyperparameter optimization. Comput Sci Discov 8(1):014008
8. Bianco MJ, Gerstoft P (2018) Travel time tomography with adaptive dictionaries. IEEE Trans Comput Imaging 4(4):499–511
9. Bishop CM et al (1995) Neural networks for pattern recognition. Oxford University Press, Oxford
10. Bisong E (2019) Google colaboratory. In: Building machine learning and deep learning models on google cloud platform. Springer, pp 59–64
11. Blitz J, Simpson G (1995) Ultrasonic methods of non-destructive testing, vol 2. Springer, Berlin
12. Bodin T, Sambridge M (2009) Seismic tomography with the reversible jump algorithm. Geophys J Int 178(3):1411–1436
13. Cao R, Earp S, de Ridder SA, Curtis A, Galetti E (2020) Near-real-time near-surface 3D seismic velocity and uncertainty models by wavefield gradiometry and neural network inversion of ambient seismic noise. Geophysics 85(1):KS13–KS27
14. Capineri L, Tattersall H, Silk M, Temple J (1992) Time-of-flight diffraction tomography for NDT applications. Ultrasonics 30(5):275–288
15. Cha YJ, Choi W, Büyüköztürk O (2017) Deep learning-based crack damage detection using convolutional neural networks. Comput Aided Civ Infrastruct Eng 32(5):361–378
16. Chassignole B, Villard D, Dubuget M, Baboux J, Guerjouma RE (2000) Characterization of austenitic stainless steel welds for ultrasonic NDT. In: AIP conference proceedings, vol 509. American Institute of Physics, pp 1325–1332
17. Dai S, Han M, Xu W, Wu Y, Gong Y, Katsaggelos AK (2009) Softcuts: a soft edge smoothness prior for color image super-resolution. IEEE Trans Image Process 18(5):969–981
18. Dukler Y, Li W, Lin A, Montúfar G (2019) Wasserstein of wasserstein loss for learning generative models. In: International conference on machine learning. PMLR, pp 1716–1725
19. Duric N, Littrup P, Babkin A, Chambers D, Azevedo S, Kalinin A, Pevzner R, Tokarev M, Holsapple E, Rama O et al (2005) Development of ultrasound tomography for breast imaging: technical assessment. Med Phys 32(5):1375–1386
20. Earp S, Curtis A (2020) Probabilistic neural network-based 2D travel-time tomography. Neural Comput Appl 32(22):17077–17095
21. Earp S, Curtis A, Zhang X, Hansteen F (2020) Probabilistic neural network tomography across Grane field (North Sea) from surface wave dispersion data. Geophys J Int 223(3):1741–1757
22. Eltony AM, Shao P, Yun SH (2020) Measuring mechanical anisotropy of the cornea with Brillouin microscopy. arXiv:2003.04344 (2020)
23. Fan Y, Xu K, Wu H, Zheng Y, Tao B (2020) Spatiotemporal modeling for nonlinear distributed thermal processes based on KL decomposition, MLP and LSTM network. IEEE Access 8:25111–25121
24. Fan Z, Mark AF, Lowe MJ, Withers PJ (2015) Nonintrusive estimation of anisotropic stiffness maps of heterogeneous steel

welds for the improvement of ultrasonic array inspection. IEEE Trans Ultrason Ferroelectr Freq Control 62(8):1530–1543

25. Guadarrama S, Dahl R, Bieber D, Norouzi M, Shlens J, Murphy K (2017) Pixcolor: pixel recursive colorization. arXiv:1705.07208 (2017)

26. Ha, V.K., Ren, J., Xu, X., Zhao, S., Xie, G., Vargas, V.M.: Deep learning based single image super-resolution: a survey. In: International conference on brain inspired cognitive systems. Springer, pp 106–119 (2018)

27. Hoffmeister BK, Verdonk ED, Wickline SA, Miller JG (1994) Effect of collagen on the anisotropy of quasi-longitudinal mode ultrasonic velocity in fibrous soft tissues: a comparison of fixed tendon and fixed myocardium. J Acoust Soc Am 96(4):1957–1964

28. Huang F, Li J, Huang, H (2021) Super-adam:faster and universal framework of adaptive gradients. arXiv:2106.08208

29. Huang H, Li Q, Zhang D (2018) Deep learning based image recognition for crack and leakage defects of metro shield tunnel. Tunn Undergr Space Technol 77:166–176

30. Isola P, Zhu JY, Zhou T, Efros AA (2017) Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 1125–1134

31. Javadi Y, MacLeod CN, Pierce SG, Gachagan A, Lines D, Mineo C, Ding J, Williams S, Vasilev M, Mohseni E et al (2019) Ultrasonic phased array inspection of a Wire+ Arc Additive Manufactured (WAAM) sample with intentionally embedded defects. Addit Manuf 29:100806

32. Javadi Y, Sweeney NE, Mohseni E, MacLeod CN, Lines D, Vasilev M, Qiu Z, Vithanage RK, Mineo C, Stratoudaki T et al (2020) In-process calibration of a non-destructive testing system used for in-process inspection of multi-pass welding. Mater Des 195:108981

33. Jiang K, Wang Z, Yi P, Wang G, Lu T, Jiang J (2019) Edge-enhanced GAN for remote sensing image superresolution. IEEE Trans Geosci Remote Sens 57(8):5799–5812

34. Jospin LV, Buntine W, Boussaid F, Laga H, Bennamoun M (2020) Hands-on Bayesian neural networks—a tutorial for deep learning users. arXiv:2007.06823 (2020)

35. Kelly B, Matthews TP, Anastasio MA (2017) Deep learning-guided image reconstruction from incomplete data. arXiv:1709.00584 (2017)

36. Keys R (1981) Cubic convolution interpolation for digital image processing. IEEE Trans Acoust Speech Signal Process 29(6):1153–1160

37. Khairi MTM, Ibrahim S, Yunus MAM, Faramarzi M, Sean GP, Pusppanathan J, Abid A (2019) Ultrasound computed tomography for material inspection: principles, design and applications. Measurement 146:490–523

38. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv:1412.6980

39. Kumar A, Sodhi SS (2020) Comparative analysis of gaussian filter, median filter and denoise autoenocoder. In: 2020 7th international conference on computing for sustainable global development (INDIACom). IEEE, pp 45–51

40. Lebedev S, Van Der Hilst RD (2008) Global upper-mantle tomography with the automated multimode inversion of surface and S-wave forms. Geophys J Int 173(2):505–518

41. Ledig C, Theis L, Huszár F, Caballero J, Cunningham A, Acosta A, Aitken A, Tejani A, Totz J, Wang Z, et al (2017) Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 4681–4690

42. Lines D (1998) Rapid inspection using integrated ultrasonic arrays. Insight Non-Destruct Test Cond Monit 40(8):573–577

43. Moya A, Irikura K (2010) Inversion of a velocity model using artificial neural networks. Comput Geosci 36(12):1474–1483

44. Nageswaran C, Carpentier C, Tse Y (2009) Microstructural quantification, modelling and array ultrasonics to improve the inspection of austenitic welds. Insight Non-Destruct Test Cond Monit 51(12):660–666

45. OnScale: 770 Marshall Street, Redwood City, CA 94063

46. Paige CC, Saunders MA (1982) LSQR: an algorithm for sparse linear equations and sparse least squares. ACM Trans Math Softw (TOMS) 8(1):43–71

47. Pyle RJ, Bevan RL, Hughes RR, Rachev RK, Ali AAS, Wilcox PD (2020) Deep learning for ultrasonic crack characterization in NDE. In: IEEE transactions on ultrasonics ferroelectrics, and frequency control

48. Rakotonirina NC, Rasoanaivo A (2020) ESRGAN+: further improving enhanced super-resolution generative adversarial network. In: ICASSP 2020–2020 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 3637–3641

49. Rawlinson N, Sambridge M (2004) Wave front evolution in strongly heterogeneous layered media using the fast marching method. Geophys J Int 156(3):631–647

50. Reddi S, Zaheer M, Sachan D, Kale S, Kumar S (2018) Adaptive methods for nonconvex optimization. In: Proceeding of 32nd conference on neural information processing systems (NIPS 2018)

51. Rodrigues TA, Duarte V, Avila JA, Santos TG, Miranda R, Oliveira J (2019) Wire and arc additive manufacturing of HSLA steel: effect of thermal cycles on microstructure and mechanical properties. Addit Manuf 27:440–450

52. Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: International conference on medical image computing and computer-assisted intervention. Springer, pp 234–241

53. Senechal M (1993) Spatial tessellations: Concepts and applications of Voronoi diagrams. Science 260(5111):1170–1173

54. Sethian JA (1999) Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science, vol 3. Cambridge University Press, Cambridge

55. Tant KM, Galetti E, Mulholland A, Curtis A, Gachagan A (2018) A transdimensional Bayesian approach to ultrasonic travel-time tomography for non-destructive testing. Inverse Prob 34(9):095002

56. Tant KMM, Galetti E, Mulholland A, Curtis A, Gachagan A (2020) Effective grain orientation mapping of complex and locally anisotropic media for improved imaging in ultrasonic non-destructive testing. Inverse Probl Sci Eng 28:1694–1718

57. Trampert J, Leveque JJ (1990) Simultaneous iterative reconstruction technique: physical interpretation based on the generalized least squares solution. J Geophys Res Solid Earth 95(B8):12553–12559

58. Virieux J, Operto S (2009) An overview of full-waveform inversion in exploration geophysics. Geophysics 74(6):WCC1–WCC26

59. Virkkunen I, Koskinen T, Jessen-Juhler O, Rinta-Aho J (2021) Augmented ultrasonic data for machine learning. J Nondestr Eval 40(1):1–11

60. Wang L, Xue J, Wang Q (2019) Correlation between arc mode, microstructure, and mechanical properties during wire arc additive manufacturing of 316L stainless steel. Mater Sci Eng A 751:183–190

61. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 13(4):600–612

62. Wirgin A (2004) The inverse crime. arXiv:math-ph/0401050

63. Yang W, Zhang X, Tian Y, Wang W, Xue JH, Liao Q (2019) Deep learning for single image super-resolution: a brief review. IEEE Trans Multimed 21(12):3106–3121

64. Yoo J, Sabir S, Heo D, Kim KH, Wahab A, Choi Y, Lee SI, Chae EY, Kim HH, Bae YM et al (2019) Deep learning diffuse optical tomography. IEEE Trans Med Imaging 39(4):877–887

65. You C, Li G, Zhang Y, Zhang X, Shan H, Li M, Ju S, Zhao Z, Zhang Z, Cong W et al (2019) CT super-resolution GAN constrained by the identical, residual, and cycle learning ensemble (GAN-CIRCLE). IEEE Trans Med Imaging 39(1):188–203

66. Zhang J, Hunter A, Drinkwater BW, Wilcox PD (2012) Monte Carlo inversion of ultrasonic array data to map anisotropic weld properties. IEEE Trans Ultrason Ferroelectr Freq Control 59(11):2487–2497

67. Zhang X, Curtis A (2020) Variational full-waveform inversion. Geophys J Int 222(1):406–411

68. Zhang X, Curtis A, Galetti E, De Ridder S (2018) 3-D Monte Carlo surface wave tomography. Geophys J Int 215(3):1644–1658

69. Zhao X, Curtis A, Zhang X (2020) Bayesian seismic tomography using normalizing flows. Earth https://doi.org/10.31223/X53K6G

70. Zhu H, Komatitsch D, Tromp J (2017) Radial anisotropy of the North American upper mantle based on adjoint tomography with USArray. Geophys J Int 211(1):349–377