Convolutional neural network analysis of x-ray diffraction data: strain profile retrieval in ion beam modified materials

MACHINE
LEARNING
Science and Technology

CrossMark

OPEN ACCESS

RECEIVED
12 September 2022

REVISED
22 November 2022

ACCEPTED FOR PUBLICATION
13 December 2022

PUBLISHED
10 January 2023

PAPER

Convolutional neural network analysis of x-ray diffraction data: strain profile retrieval in ion beam modified materials

A Boulle[1,*] and A Debelle[2]

[1] IRCER, CNRS UMR 7315, Limoges, France
[2] IJCLab, Université Paris-Saclay, CNRS/IN2P3, Orsay, France
[*] Author to whom any correspondence should be addressed.

E-mail: alexandre.boulle@cnrs.fr

Keywords: x-ray diffraction, strain, convolutional neural networks, machine learning, ion irradiation, ion implantation

Abstract

This work describes a proof of concept demonstrating that convolutional neural networks (CNNs) can be used to invert x-ray diffraction (XRD) data, so as to, for instance, retrieve depth-resolved strain profiles. The determination of strain distributions in disordered materials is critical in several technological domains, such as the semiconductor industry for instance. Using numerically generated data, a dedicated CNN has been developed, optimized, and trained, with the ultimate objective of inferring spatial strain profiles on the sole basis of XRD data, without the need of *a priori* knowledge or human intervention. With the example ZrO_2 single crystals, in which atomic disorder and strain are introduced by means of ion irradiation, we investigate the physical parameters of the disordered material that condition the performances of the CNN. Simple descriptors of the strain distribution, such as the maximum strain and the strained depth, are predicted with accuracies of 94% and 91%, respectively. The exact shape of the strain distribution is predicted with a 82% accuracy, and 76% for strain levels $<2\%$ where the amount of meaningful information in the XRD data is significantly decreased. The robustness of the CNN against the number of predicted parameters and the size of the training dataset, as well as the uniqueness of the solution in some challenging cases, are critically discussed. Finally, the potential of the CNN has been tested on real, experimental, data. Interestingly, while the CNN has not been trained to operate on experimental data, it still shows promising performances with predictions achieved in a few seconds and corresponding root-mean-square errors in the 0.12–0.17 range for a fully automated approach, *vs.* a 0.06–0.12 range for a classical, human-based, approach that, in turn, requires several tens of minutes to optimize the solution. While the overall accuracy of the CNN has to be improved, these results pave the way for a fully automated XRD data analysis.

1. Introduction

X-ray diffraction (XRD) is an interferometric phenomenon which results from the coherent superposition of x-ray waves scattered by individual atoms when a material is illuminated with x-ray radiation. Within the ‘kinematical theory’ of diffraction [1] the amplitude of the electric field of the diffracted x-rays is given by the sum:

$$E(\mathbf{Q}) = \sum_j f_j \exp\left(i\,\mathbf{Q}\cdot\mathbf{r}_j\right) \tag{1}$$

where f_j is the amplitude scattered by the jth atom in the material, and the scalar product $\mathbf{Q}\cdot\mathbf{r}_j$ is the phase difference between each scattered wave, $\mathbf{Q}$ and $\mathbf{r}_j$ being the wave vector transfer (also referred to as the scattering vector) and the coordinate vector of the jth atom, respectively. The amplitude of $\mathbf{Q}$ is equal to 4π $\sin\theta/\lambda$, where θ is half the diffraction angle and λ is the x-ray wavelength. The diffracted amplitude is extremely sensitive to, even minute, variations in the atomic coordinates, which explains why XRD has been used since decades to study the structure of crystalline materials, their nano- and microstructure and crystalline defects [1–4]. However, the quantity measured in an actual XRD experiment is the intensity, i.e. the squared modulus of the amplitude, $E \times E^*$; the phase of the amplitude is therefore lost in the process, which gives rise to the so-called 'phase problem', common to all diffraction techniques, which implies that only coordinate-differences can be obtained via a direct inversion of the intensity. To circumvent this issue, scientists usually rely on modelling and fitting algorithms where a parameterized physical model is iteratively adjusted until a satisfactory agreement with the experimental data is obtained[3]. It should be noted that, for the sake of simplicity, we rely on the kinematical theory of diffraction to present the phase problem, whereas all calculations in the following are performed using the more accurate 'dynamical theory' of diffraction. However, as far as the phase problem is concerned, the conclusions drawn using the kinematical theory remain perfectly valid within the dynamical theory.

One important application of XRD is the determination of the spatial distribution of strain in disordered materials, like for instance, those submitted to ion irradiation. Being able to precisely characterize these materials is of critical importance in many fields, for instance to understand strain and defect formation during ion implantation in the semi-conductor industry [5], a family of materials for which XRD is particularly well adapted [6]. This aspect is especially important in this period of time, where increasing the semiconductor supply chain resilience against international crises is now a strategic priority in Europe and the United States of America. For instance, in the USA, the recent CHIPS and Science act[4] will allocate $280 billion in the next 5 years to the semiconductor industry, among which $174 billion will be devoted to investments in Science, Technology, Engineering and Math programs, workforce development, and research & development. A similar, though less ambitious, €43 billion program will be announced in Europe as well[5]. In this context, the development of high precision in-line metrology systems, while maintaining high throughput, is an important challenge to face, and machine learning-based algorithms are here expected to play a crucial role.

The determination of spatial strain profiles in strained materials relies, as mentioned above, on the simulation of experimental XRD data with physical models [7–14]. These simulation methods are in general based on classic least-squares optimization algorithms which require a good guess of the initial parameters defining the strain profile in order to avoid being trapped in a local minimum of the least-square error hypersurface. Obtaining a physically sound guess may often require several tens of minutes, up to several hours, depending on the complexity of the problem. In any case, it requires *a priori* knowledge about the system, and the supervision of an expert, in order to set constraints on the fitting parameters during the optimization procedure, so as to avoid the emergence of unphysical solutions. Global search algorithms such as, for instance, simulated annealing [15], may solve the local minimum issue but those are intrinsically slower than usual local minimization techniques.

Machine leaning algorithms, and in particular artificial neural networks (NNs), can be used to solve that type of inverse problems, thereby considerably decreasing computing times and eliminating the need of *a priori* knowledge. The general idea is to map reciprocal space features (obtained by diffraction) to real space features (here, the strain profile), by feeding XRD data into a NN and adjusting its parameters so that its output matches the expected (strain profile) result; a procedure referred to as 'training' the NN. In the recent years, NNs have been increasingly used in the field of the characterization of materials using the scattering of x-rays or neutrons. One can cite symmetry identification [16–18], space group identification [19–23], phase identification [24–27], phase and crystallite size quantification [28], lattice parameter prediction [29], classification of 2D diffraction data in single particle imaging [30, 31], or in small and wide angle scattering data [32–41], data correction [42–44] or automated analysis of 2D diffraction images [45–49], and phase retrieval with synchrotron coherent x-rays [50–55].

The studies which are more closely related to the present work are those dealing with the inversion of neutron or x-ray reflectivity data recorded from thin films or multilayers, so as to retrieve physical parameters such as the film thickness and surface/interface roughness [56–60]. These parameters then allow the authors to reconstruct the, so-called, scattering length density profile. In these studies the number of layers constituting the multilayer structure were limited to 2 [56–59], and up to 3 [60], with impressive performances on actual samples. Arbitrary profiles have been considered in [61] but without any application to actual samples. Although there is a clear methodological similarity between the inversion of neutron/x-ray reflectivity curves and XRD curves, the latter differs from reflectivity in many aspects:

[3] In this work we do not consider 'phase retrieval' techniques that are used to image individual nanoparticles (and their strain distribution) and which rely on coherent x-rays that can only be delivered by synchrotron sources, see, e.g. [83].

[4] https://www.congress.gov/117/plaws/publ167/PLAW-117publ167.pdf

[5] https://ec.europa.eu/info/strategy/priorities-2019-2024/europe-fit-digital-age/european-chips-act_en

- Ion-irradiated or ion-implanted materials in general display both strain and disorder, where the former is related to long range variations of interatomic distances, and the latter refers to random atomic displacements. XRD is therefore affected by the lattice strain profile and the random atomic disorder profile (via the so-called Debye–Waller -*DW*- factor), whereas reflectivity solely depends on the scattering length density profile;
- The strain and DW profiles can exhibit any arbitrary shape, whereas the scattering length density profiles are, to a certain extent, constrained by the multilayer geometry;
- The wave vector transfer range depends on the level of strain (see below), so that the data range is not constant (which is an important condition for the use of NNs).

The goal of this work is to provide a proof of concept that NNs can be used to retrieve strain profiles in disordered crystals, thereby considerably decreasing the computing time and, ultimately, removing the necessity of *a priori* knowledge. To the best of our knowledge, we are not aware of any similar attempt in published studies. In the present work we focus of the feasibility of such an approach and examine the parameters that condition the NN performances. It is important to mention that, although the applicability of the method is tested on real data, we do not attempt to provide detailed benchmarks of the NNs *vs.* fitting approaches. Such aspects will be considered in future works.

2. Training data and NN architecture

2.1. Training data generation

It is not unusual for NNs to contain up to several 10^5–10^6 parameters that have to be adjusted. A large number of input data, of, roughly, the same order of magnitude, is therefore required for the training to be successful. Acquiring and analysing 10^6 XRD curves being obviously impossible, these data have been numerically generated, similarly to other studies [56–61]. For this purpose, we relied on a theoretical framework described in details in the past [11]. In short, the XRD curves were computed within the framework of the dynamical theory of diffraction, using the recursive solution [62] to the Takagi–Taupin differential equations [63, 64].

Ion-irradiated crystals can be described as a system composed of a film (i.e. the irradiated region) on a substrate (i.e. the unirradiated, pristine, region), both sharing the same structure, orientation and composition. Within the film, the level of strain and disorder vary as a function of depth, while these two parameters remain unchanged in the pristine part of the crystal. Using the previously described theoretical framework, an XRD curve can be generated when provided with a given strain and disorder depth-profiles.

2.1.1. Strain and Debye–Waller depth-profiles

Strain depth-profiles in ion-irradiated materials can exhibit a wide variety of shapes, ranging from purely decreasing strain from the surface to the interface, to profiles with a more or less pronounced maximum within the irradiated region or even almost constant profiles, or any combination of those. In all cases, the strain reaches 0 at the film/substrate interface. In order to produce realistic strain depth-profiles, we followed a three step procedure. In a first step, we use an asymmetrical Gaussian function to produce a peak-shaped function. However, actual disordered materials in general exhibit strain profiles that can not be described with such a simple analytical function. This is why, in a second step, in order to add some realistic deviation from simple shapes, we add a constant value on either or both sides of the peak. Finally, in a third step, local random variations of the strain profiles are introduced using the *B*-spline parameterization we introduced in our previous work [65]. The details of the procedure are given below.

In the first step, the strain profile is generated with an asymmetric Gaussian function, normalized to unit maximum:

$$G(z) = \exp\left(-\frac{1}{2}\left[\frac{z-\mu}{\sigma}\right]^2\right) \qquad (2)$$

where z is the depth coordinate from the surface, μ is the position of the maximum and σ is the standard deviation obeying the following condition: $\sigma = \sigma_L$ if $z < \mu$, $\sigma = \sigma_R$ otherwise. The parameters μ and σ_R are randomly sampled in the $[0, t]$ range, where t is the disordered (i.e. film) thickness with the constraint $\mu = t - 3 \times \sigma_R$ in order to ensure that the strain converges to 0 at the film/substrate interface, whereas σ_L is randomly sampled in the $[0, \infty)$ range. The thickness is randomly selected in the [50 nm, 1000 nm] range (see table 1), those values corresponding to the damaged thickness usually observed in materials subjected to ion-induced ballistic damage. In the second step, a constant value (randomly selected within the [0, 0.8] range) is added on the $z < \mu$, $z > \mu$, or both, sides with an equal probability of 0.25.

Table 1. Range of parameters used for the data generation of the different CNNs used in this work.

	t (nm)	$e_{\max}$ (%)	DW	H (deg.)	background	$I_{\max}$	DR
CNN1	50–1000	0.2–8	0.001–1	0.003–0.1	1–100	10^3–10^7	3–6
CNN2	100–1000	0.5–8	0.05–1	0.003–0.1	1–100	10^3–10^7	3–6
CNN3	100–1000	0.5–2	0.05–1	0.003–0.1	1–100	10^3–10^7	3–6

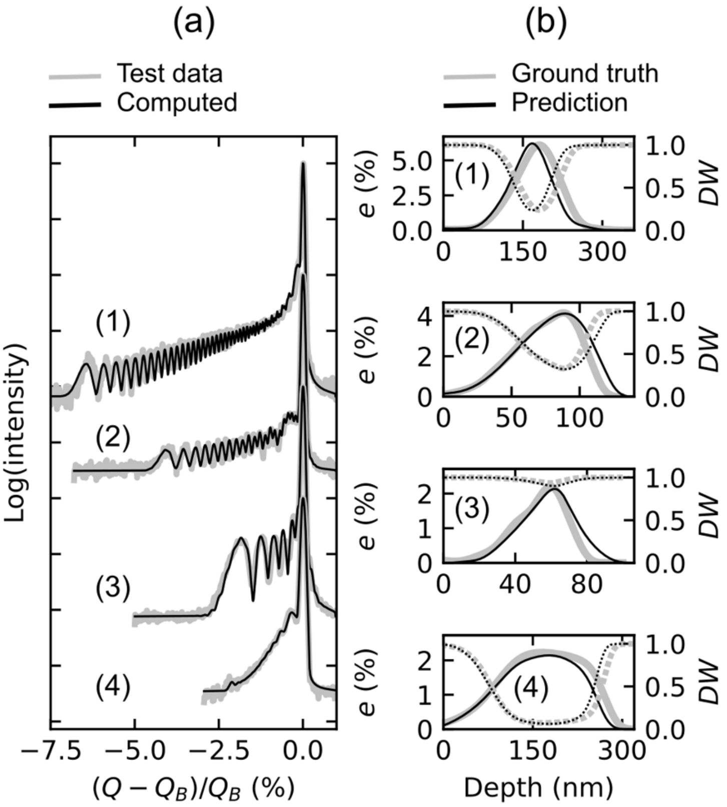

Figure 1. (a) Generated test XRD profiles (grey lines) and XRD profiles computed from the prediction of the neural network (black lines). (b) Ground truth strain profiles (continuous grey lines) and DW profiles (dotted grey lines); predicted strain profiles (continuous black lines) and DW profiles (dotted black lines). The XRD curves labelled (1–4) in (a) correspond the strain/DW profiles labelled (1–4) in (b).

In the last step, the generated strain profile is fitted with a *B*-spline function defined as:

$$e(z) = e_{\max} \times \sum_{i=1}^{N_w} w_i B_{i,3}(z) \tag{3}$$

where $e(z)$, $e_{\max}$, w_i and $B_{i,3}$ are the strain profile, the maximum strain value, the weight of the ith *B*-spline function and the ith 3rd degree *B*-spline basis function. Complete details regarding the *B*-spline description of strain are given in [65]. The *B*-spline parameterization is a convenient way to describe complex, arbitrarily shaped functions, while requiring a limited set of parameters. In the present case we chose 10 *B*-spline basis functions to describe the strain profile, which corresponds to 10 adjustable weights w_i. The fitted weights w_i are then allowed to randomly deviate from their initial value within a ± 10% range, with a uniform probability, to introduce randomness in the overall shape. Finally, the maximum strain value, $e_{\max}$, is chosen in the [0.2%, 8%] range with uniform probability (see table 1), the 8% value roughly corresponding to the maximum strain reported experimentally [66–68]. Examples of generated strain profiles are given in figure 1(b) as grey lines.

The effect of random atomic displacement is described by the Debye–Waller factor, which is a function of the root-mean-square (rms) atomic displacements [1]. The DW factor is equal to 1 in a perfect crystal and reaches 0 for heavily disordered crystals. Although the shape of the DW profile might significantly differ from the shape predicted from the strain profile [11], for simplicity we shall here assume that it is given by the following equation [69]:

$$\mathrm{DW}\left(z\right) = \exp\left\{-\alpha \times \left[e\left(z\right)/e_{\max}\right]^2\right\} \tag{4}$$

where $\alpha = 0$ corresponds to a perfect crystal and $\alpha \gg 1$ correspond to a heavily disordered crystal. The parameter α is randomly selected so that the resulting minimum value of DW(z), $\mathrm{DW}_{\min}$, uniformly lies in the [0.001, 1] range, see table 1. Examples of generated DW profiles are given in figure 1(b) as dotted grey lines. It should be noted that we restrict ourselves to tensile strain, as compressive strain is almost never encountered in irradiated materials.

2.1.2. XRD training data

With the previous strain and DW profiles we compute the diffracted intensity (I) distribution as described earlier. For the calculations, we considered, as an example, the 400 reflection of cubic Y_2O_3-doped ZrO_2 (9.5% Y_2O_3) with a lattice parameter $a = 5.145$ Å. The magnitude of the scattering vector for the 400 Bragg position is $Q_B = 4.885$ Å^{-1}.

The following procedure has been used to generate the XRD data:

- Using the strain and DW profiles as input, the intensity has been computed in a range of scattering angles [$\theta_{\min}$, $\theta_{\max}$] with a constant step of 0.005°. The limits, $\theta_{\min}$ and $\theta_{\max}$, were randomly chosen, but wide enough to include the Bragg peak from the perfect crystal and the signal from the disordered region. The computed intensity has been normalized to unity.
- The resulting intensity has then been convoluted with a Gaussian function with full-width at half-maximum (H) randomly selected in the [0.003°, 1°] range. This allowed to account for the broadening induced by the diffractometer (limited resolution due to beam divergence or spectral dispersion), or by the quality of the unirradiated crystal which might exhibit deviations from perfection.
- In order to account for the actual intensity encountered on most laboratory diffractometers, the computed intensity has been multiplied by a scale factor 10^κ with κ selected in the [3, 7] range with uniform probability. A constant background 10^ν, due to diffuse scattering from the sample or from imperfect optics in the diffractometer, was also added, with ν selected in the [0, 2] range with uniform probability. Finally, in order to mimic data usually recorded on laboratory diffractometer, all data with a dynamic range, defined as $\mathrm{DR} = \log(I_{\max}/I_{\min})$, outside the [3, 6] range were rejected ($I_{\min}$ and $I_{\max}$ correspond to the minimum and maximum values of the intensity *in a given XRD curve*).
- In order to account for counting statistics, the computed intensity I has been replaced with random numbers $k \times I$, where k obeys the Poisson distribution $P(k; I)$.
- Finally, the angular θ range has been converted to scattering vectors using $Q = 4\pi\ \sin\theta/\lambda$.

All parameter ranges used for the strain, DW and intensity calculations are summarized in table 1 (row labelled 'CNN1'). Examples of generated XRD data are given in figure 1(a), grey lines. All curves exhibit common features:

- An intense and narrow peak located at $\Delta Q/Q_B = (Q - Q_B)/Q_B = 0$, which corresponds to the Bragg diffraction from the perfect crystal below the disordered region (i.e. the substrate).
- An additional signal at $\Delta Q/Q_B < 0$ exhibiting more or less visible interference fringes. This intensity emanates from the disordered region (i.e. the film), the extension of which is proportional to the level of strain, $(\Delta Q/Q_B)_{\max} \approx -e_{\max}$. The presence of fringes results from the interferences of x-ray waves scattered from regions with the same level of strain. It is the presence of those fringes that allows for an accurate determination of the strain and DW profiles.
- The intensity of the additional signal as compared to the perfect crystal is modulated by both the DW factor and the thickness: a low DW factor and/or a low thickness result in a lower intensity.

The fact that the non-zero intensity range depends on the level of strain results in data ranges of varying length, at least in actual experiments, where there is no added value in recording empty regions. We decided to replicate this feature, which has two consequences:

- In order to bring all the data ranges to the same length, missing values were replaced with constant values corresponding to the minimum value observed in the initial data.
- This prevented us to use dense neural networks (DNNs) as those are sensitive to the absolute values of the pixels in the data. We experienced that introducing data with arbitrary values prevented the DNNs to be correctly trained, resulting in incorrect predictions. This is the reason why we used convolutional neural networks (CNNs) [70] in the following, instead of DNNs that are often used to invert reflectivity data [56–59].

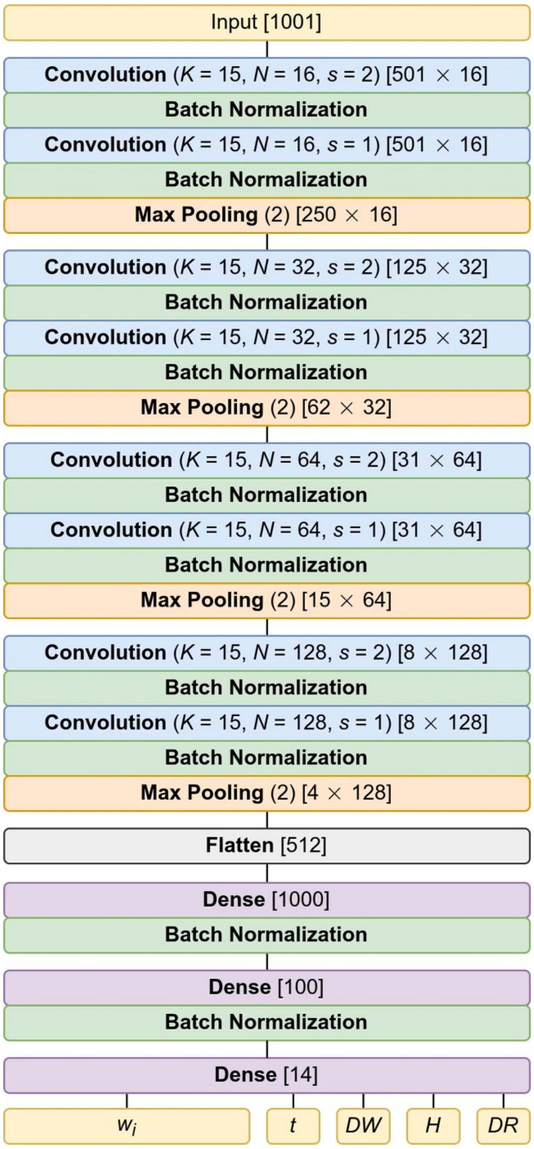
Input [1001]
Convolution (K = 15, N = 16, s = 2) [501 × 16]
Batch Normalization
Convolution (K = 15, N = 16, s = 1) [501 × 16]
Batch Normalization
Max Pooling (2) [250 × 16]
Convolution (K = 15, N = 32, s = 2) [125 × 32]
Batch Normalization
Convolution (K = 15, N = 32, s = 1) [125 × 32]
Batch Normalization
Max Pooling (2) [62 × 32]
Convolution (K = 15, N = 64, s = 2) [31 × 64]
Batch Normalization
Convolution (K = 15, N = 64, s = 1) [31 × 64]
Batch Normalization
Max Pooling (2) [15 × 64]
Convolution (K = 15, N = 128, s = 2) [8 × 128]
Batch Normalization
Convolution (K = 15, N = 128, s = 1) [8 × 128]
Batch Normalization
Max Pooling (2) [4 × 128]
Flatten [512]
Dense [1000]
Batch Normalization
Dense [100]
Batch Normalization
Dense [14]
w_i
t
DW
H
DR

Figure 2. Architecture of the CNN. The role of each layer is indicated in the corresponding box. For convolution layers, K, N and s respectively correspond to the kernel size in pixels, the number of convolution kernels and the stride used for the convolution. Each Max Pooling operation is performed over 2 pixels. The size of the output tensor of each layer is indicated in square brackets. The input vector corresponds to the 1001 intensity values. The output vector is composed of the 10 *B*-spline weights (w_i), the thickness of the damaged region, *t*, the DW factor, the width of the Bragg peak, *H*, and the dynamic range, DR.

We generated 1.5×10^6 XRD patterns (each consisting in a vector $\mathbf{x}$ with 1001 intensity values), 10% of which were removed from the training set and used as a test set, i.e. to check the performances of the NN on never-seen-before data. The choice of the data set size is justified in section 3.5 below. It is known that during NN training, it is preferable to compress the data into a limited magnitude range, otherwise the largest values would bias the training. To do so, we transformed the intensity as follows: $\hat{I} = \log(I/I_{\max})$. As a result, the training and test data lie in the [0, −DR] range. The generation of 1.5×10^6 XRD patterns requires 2.5 h with a workload parallelized over 96 CPU cores.

2.2. Network architecture and training

2.2.1. Network architecture

The CNNs have been developed and trained with the Python *TensorFlow* (www.tensorflow.org/) library [71] using the *Keras* (https://keras.io/) front-end [72]. The CNN we developed in this work is displayed in figure 2 and contains $\sim 1.104 \times 10^6$ trainable parameters. There are no definite rules for designing the architecture of a NN, and those essentially derive from empirical knowledge associated to a particular problem. Despite this, we underline below some characteristics that turned out to play an important role in the performances of the CNN. All the aspects discussed below are natively included in *Keras* and do not require additional coding.

Firstly, the dilatation between the flatten layer (which transforms the 4 × 128 tensor into a linear 512 vector) and the first dense layer (1000 pixels), figure 2, was essential to reach convergence. Without this dilatation, no learning was possible. Dense layers (also referred to as fully connected layers) are defined such that the output of the jth neuron of the lth layer, $a^{l}{}_{j}$, is a linear function (with weights $w^{l}{}_{jk}$ and bias $b^{l}{}_{j}$) of the output of all neurons of the preceding layers passed through an activation function σ [73]:

$$a_j^l = \sigma\left(\sum_k w_{jk}^l a_k^{l-1} + b_j^l\right). \tag{5}$$

The weights and biases are trainable parameters. In all layers of the CNN (except the very last dense layer), a rectified linear unit was used as an activation function, $\mathrm{ReLU}(x) = \max(0, x)$. Other activation functions have been tested, such as the leaky-ReLU and the parametric-ReLU, without visible improvement.

Secondly, we used a relatively large convolution kernel size (also referred to as a receptive field), $K = 15$ pixels, as compared to the small sizes, e.g. 3 × 3 pixels, used in image recognition. This kernel size turned out to be more well suited to XRD data where there are no sharp and well defined edges, but rather continuous intensity variations. N of such kernels were used, which produces N different output vectors (also referred to as feature maps) of size M/s where M is the size of the vector of the preceding layer, and s is the stride (i.e. the step used in the convolution operation)[6]. The jth component in the mth feature map of the lth layers writes [73]:

$$a_{j,m}^l = \sigma\left(\sum_n \sum_k w_{k,n}^l a_{s\times j+k,n}^{l-1} + b_n^l\right) \tag{6}$$

where $w^{l}{}_{k,n}$ is the kth pixel value of the kernel in the nth feature map of the lth layer of the CNN, and $b^{l}{}_{n}$ is the bias in the nth feature map of the lth layer. It can be observed that the weight and bias do not depend on the position (j) of the output neuron, i.e. the same weights and biases are shared across the whole input data. This makes the specificity of convolutional layers, which detects translation-invariant features, while requiring much less training parameters [74]. The so-called 'Max Pooling' layers aim at merging similar feature detected in the input vector by selecting the maximum value within a given pixel range (here, 2 pixels), thereby reducing the vector size by a factor of 2.

Thirdly, the use of two successive convolution layers (first introduced in [75] and then widely re-used [76]) have a similar effect than an increase in the kernel size, but with a reduced number of parameters and increased feature detection thanks to successive activation functions.

Finally, the batch normalization layers [77] allowed to reach a deeper minimum of the cost function (see below) during the training, with a reduced number of training epochs (see below for definition). The output of the batch normalization layer is

$$\begin{aligned} z_i &= \gamma \hat{x}_i + \beta \\ \text{with}\, \hat{x}_i &= \tfrac{x_i - \langle x_i \rangle}{\sqrt{\langle x_i^2 \rangle}} \end{aligned} \tag{7}$$

where x_i is the ith component of the input vector, γ and β are trainable parameters. The mean and standard deviations are evaluated over the training data set of a given training step (a mini-batch, see below).

The output of the CNN is a vector, $\mathbf{y}$, with the following components: the 10 B-spline weights (not to be confused with the weights of the NN), the disordered thickness, the minimum DW factor, the width of the convolution Gaussian and the dynamic range. The two latter parameters are not dependent on the material but rather on the diffractometer resolution or sample quality, and the measurement conditions, respectively. They are nonetheless included in the NN so as to allow the calculation of the diffracted intensity on the sole basis of the NN's output. To avoid issues due to scale differences between these variables, the expected outputs (ground truth) are rescaled to a [0–1] range using

$$\hat{y}_i = \frac{y_i - \min(y_i)}{\max(y_i) - \min(y_i)} \tag{8}$$

where y_i is the ith component of the ground truth vector, and the minimum and maximum values are evaluated *over the whole training data set*.

[6] We used the 'padding' option in the convolution operation implemented in *TensorFlow*. This allows to avoid edge effects and keep the same vector/tensor size before and after convolution (when using a stride of 1). Without this option the vector size is reduced by K-1 pixels.

2.2.2. Training procedure

The CNN is trained by minimizing the mean-square-error (mse):

$$C(\mathbf{p}) = \frac{1}{N_\mathrm{d}} \sum_{i=1}^{N_\mathrm{d}} \|f(\mathbf{x}_i;\mathbf{p}) - \hat{y}_i\|^2 \tag{9}$$

where $f(\mathbf{x}_i;\mathbf{p})$ is the response vector of the CNN for an input data vector $\mathbf{x}_i$ and a set of trainable parameters $\mathbf{p}$, and N_d is the number of data points used for the evaluation of the gradient. The minimization is carried out by iteratively updating the parameters $\mathbf{p}$ in the direction of descending gradient of the cost function C, $\nabla_\mathbf{p}(C)$. For this purpose, we used the usual mini-batch stochastic gradient descent algorithm, where subsets of the initial training data (mini-batches, of size N_d) are randomly selected and used to evaluate the gradient [72]. We used $N_\mathrm{d} = 2096$ as mini-batch size. The gradient itself is calculated with the famous back-propagation algorithm [70, 78]. The parameter optimization is performed with the RMSprop algorithm [72, 79] implemented in Keras. The learning rate, i.e. the strength with which the parameters are updated, starts at the usual value of 0.001 and decreases by a factor of 10 when the cost function stagnates for at least 10 epochs. An epoch corresponds to a situation where all data points have been used once to evaluate the gradient. Alternative optimizers, such as the widespread Adam algorithm [80], have been tested but did not bring any noticeable improvement as compared to RMSprop.

In addition to the 10% data reserved for testing (1.5×10^5), another 10% is used for validation (i.e. to tune the parameters of the CNN using data not used for the training). This leaves 1.215×10^6 training data and 1.35×10^5 for validation. The CNN was trained over 70 epochs. With the hardware used in this work (Nvidia RTX A6000 GPU) the training lasted 13 min (11 sec/epoch).

3. Results and discussion

3.1. Predictions, errors and accuracy

A typical learning curve is given in the supplementary material (figure S1). The mse remains constant after 65–70 epochs with a value of 0.001. The mse evaluated on the validation set reaches 0.0016, and no sign of over-fitting (i.e. a degradation of the validation mse) could be detected, which is likely a consequence of the large amount of data used for training.

To evaluate the performances of the CNN, the test data set (with 150 000 XRD curves) has been used to predict the different parameters considered in this work which, we recall, are the thickness of the strained region, t, the maximum strain, e_max, the minimum DW factor, DW_min, the width of the Bragg peak, H, and the dynamic range, DR. The B-splines weights, which are the parameters describing the shape of the strain profile (equation (3)), will be discussed separately. For each of these five parameters, p, we plotted the predicted value, $p^{(p)}$, *vs.* the ground truth, figures 3(a)–(e). In each case, it appears that, overall, the values predicted by the CNN are linearly correlated with the expected values, with a dispersion depending on the parameter considered. The histogram of relative errors for each parameter p, $\Delta p/p = (p^{(\mathrm{p})}–p)/p$, is displayed in figure 3(f) as a violin plot. The parameters e_max, H and DR have a relatively low dispersion of errors, indicating that they are accurately predicted by the CNN. This can be understood by the fact that they depend on relatively simple features of the XRD data, namely the extension of the additional region in the region with $\Delta Q/Q_\mathrm{B} < 0$, the width of the intense peak at $\Delta Q/Q_\mathrm{B} = 0$, and the ratio between the maximum and minimum intensity, respectively. On the other hand the thickness and the DW factor, which both depend on more subtle features (the intensity of the additional signal and the fine structure of the fringes) are characterized by higher errors, that yet remain reasonable.

In order to better quantify the performances of the CNN, we define the prediction accuracy as follows. From the relative error histograms (figure 3(f)) we computed the cumulated distribution of absolute relative errors, $|\Delta p|/p$, and set a threshold at 0.1, figures 3(g)–(k). The corresponding fraction, a, is used as a measure of the prediction accuracy; i.e. a % of the data have a relative error lower than 0.1. The accuracy corresponding to each parameter is listed in table 2 (row labelled 'CNN1'). As already inferred from the violin plot, the parameters e_max, H and DR have the highest accuracies ($\geqslant$90%). Conversely DW and t have accuracies of 73% and 85%, respectively.

Let us now consider the most important prediction of the CNN, namely the complete strain profile, which depends not only on the B-spline weights, but also on the thickness t and the maximum strain e_max. Being a function of several parameters, the prediction can not be plotted *vs.* the ground truth. Instead, we compute the root-mean-square error (rmse), between the predicted strain profile and the ground truth, $\sigma_e = \sqrt{\langle \left[e^{(p)}(z) - e(z)\right]^2 \rangle}$, where the average is performed over all z values (i.e. over the whole strain profile) and we normalize it to the maximum strain. The corresponding plot is shown in figure 3(l) and the

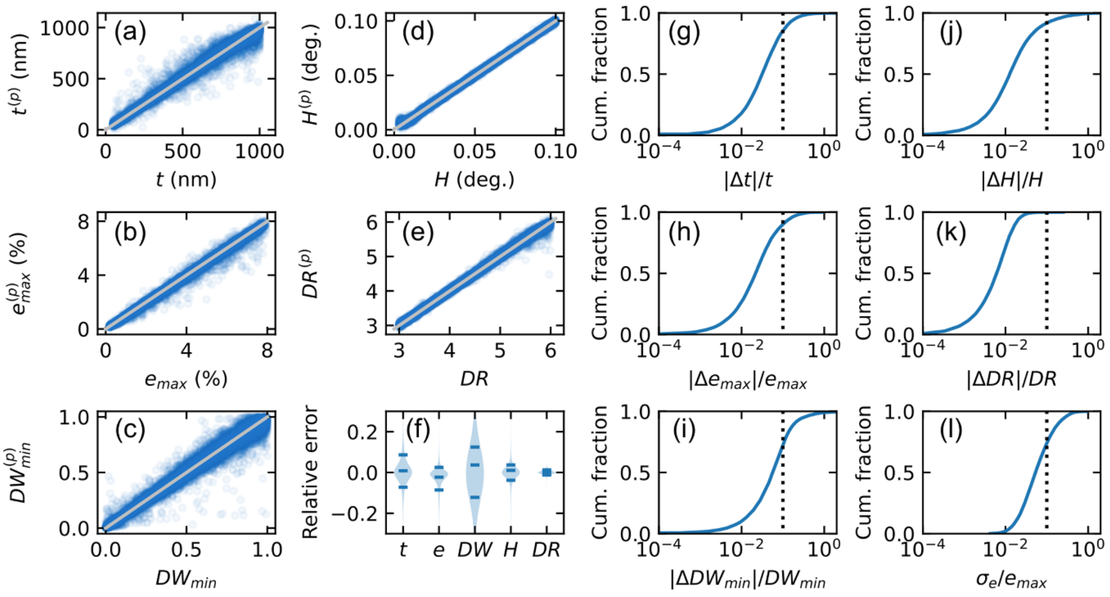

Figure 3. (a)–(e) Plot of the values predicted by the CNN (with superscript (p)) vs. the ground truth, for all parameters but the B-spline weights. (f) Violin plots showing the distribution of relative errors for each parameter. The 10th, 50th and 90th percentiles are indicated by lines. (g–k) cumulative fraction of the relative errors for each parameter. (l) Cumulative fraction of the relative root-mean-square error of the strain profile. For plotting purposes the number of data points has been reduced to 15 000, instead of the 150 000 contained in the test set.

Table 2. Summary of the train and test mse and the prediction accuracy, a, of each parameters for the different CNNs considered in this work.

	train mse	test mse	$a(t)$	$a(e_{\max})$	a(DW)	$a(H)$	a(DR)	a(strain)
CNN1	0.0010	0.0016	85%	90%	73%	92%	100%	74%
CNN2	0.00077	0.0013	91%	94%	90%	93%	100%	82%
CNN1 ($e_{\max}$ ⩽2%)			70%	74%	63%	91%	100%	50%
CNN2 ($e_{\max}$ ⩽2%)			80%	81%	81%	93%	100%	61%
CNN3	0.0019	0.0026	86%	94%	85%	91%	100%	76%
CNN3[12]	0.0022	0.0031	86%	93%	84%			76%
CNN3[3]	0.0011	0.0015	88%	97%	90%			

associated accuracy is 74%. In the following, since this work is focused on determining the strain profiles, the relative strain rmse, $\sigma_e/e_{\max}$, will be used as the unique metric to quantify the CNN performances.

Example of predicted strain profiles with a relative rmse of 0.1 are shown in figure 1(b), black lines. It can be observed that the predictions with a 0.1 relative error correspond to remarkably good predictions as compared to the ground truth. Another way of assessing the quality of the prediction is to compute the corresponding XRD curve on the basis of the predicted strain profiles. These computed curves are shown in figure 1(a). Again, it can be observed that the agreement with the test data is close to perfect. This approach will be particularly useful when real data will be considered and where no ground truth is available.

Besides the example just discussed, it can also be useful to consider two other extreme cases: those exhibiting the lowest and highest relative strain rmse, with values <0.005 and >0.3, respectively. The corresponding XRD curves and strain profiles are given in figure 4. In the case of low errors (a, b) it can be seen that the predictions of the strain profile perfectly match the ground truth. The agreement is slightly less perfect for the DW factor, which is not surprising because we here consider the strain rmse to which the DW factor does not contribute. It can be noticed that all data exhibit rather high thickness (>300 nm) and high maximum strain (>3.5%). They also exhibit rather high $\mathrm{DW}_{\min}$ values and a high dynamic range (>4). The high strain value results in large extension of the signal emanating from the strained region, whereas the large thickness, the high DW factor and the high dynamic range factors ensure a high signal-to-noise ratio in this region, hence maximizing the amount of information the CNN can rely on. Conversely in the case of high errors (figures 4(c) and (d)), the ground truth strain profiles exhibit either very low strain, very low DW factor or very low thickness. Combined with a low dynamic range (<4), there is almost no information in the XRD data. This can be confirmed by the fact that, although the predictions are completely wrong, the agreement between the computed and test XRD data is very good.

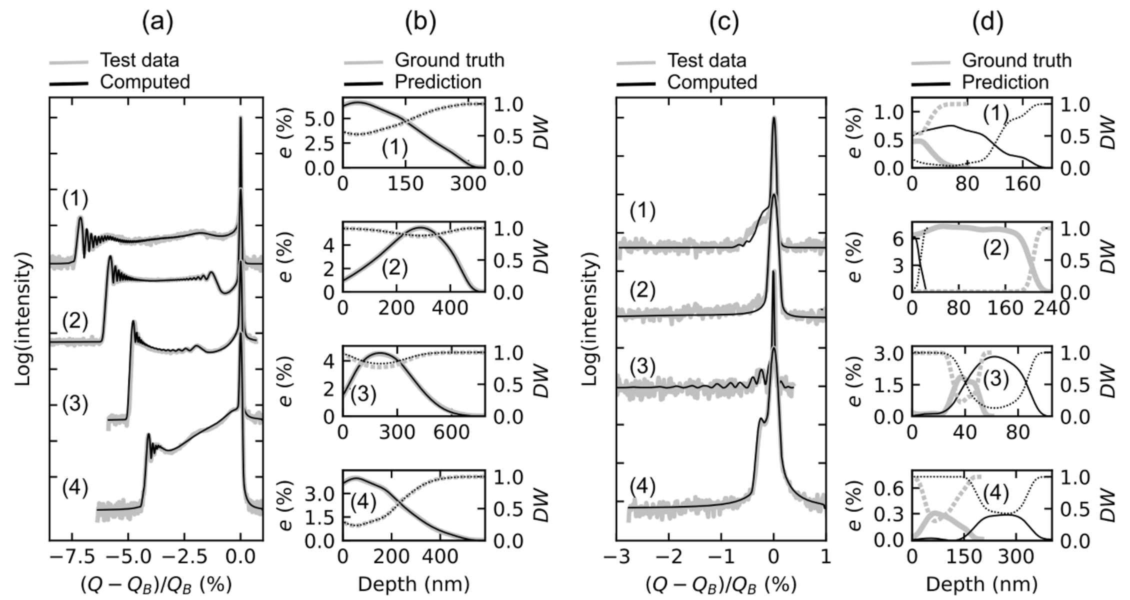

Figure 4. XRD test data (grey curves) and computed from predictions (black curves) in the case of $\sigma_e/e_{\max} < 0.005$ (a) and >0.3 (c). (b), (d) Corresponding ground truth (grey curves) and predicted strain profiles (black curves). The dashed curve correspond to the DW profiles.

3.2. Role of physical parameters on the CNN performances

In order to get a deeper insight into the influence of each parameter (strain, thickness, DW, *etc*) on the accuracy of the CNN, it can be useful to plot the relative strain rmse as a function of the ground truth value of each parameter. Such plots are given in the supplementary materials (figure S2). Although some tendencies can be detected in these plots, it is not straightforward to deduce any quantitative information. We therefore suggest the following procedure: for each parameter we select the 10^3 datasets with the lowest $\sigma_e/e_{\max}$ values and compute the histogram of the values taken by each parameter. We perform the same operation on the 10^3 datasets exhibiting the highest $\sigma_e/e_{\max}$ values. The corresponding histograms are given in figure 5.

The first observation that can be made is that the distribution of parameters in the data with the lowest errors (figures 5(a)–(e)) is in general more uniform than in the case of the data with the highest errors, where narrow peaks appear in certain ranges of parameters (figures 5(f)–(h)). This demonstrates that, while it is possible to get good performances over a rather broad range of parameters, some narrow ranges of parameters catastrophically deteriorate the CNN performances. A careful examination of figures 5(f)–(h)) reveals that these parameters are thickness $t < 100$ nm, $e_{\max} < 0.5\%$ and DW < 0.05, which confirms the qualitative observations made in the previous section: these ranges of parameters do not contain enough information in the XRD data to allow the CNN to learn features from those data. Finally the width of Bragg peak H has an weaker effect, with slightly better performances in the mid-range of the parameter, i.e. in the 0.025–0.075° range (figure 5(i)). The effect of the dynamic range, although not negligible, is much less pronounced (figure 5(j)).

Conversely, for the lowest errors (mainly figures 5(a)–(c)) it appears, unsurprisingly, that large values of t, $e_{\max}$ and DW, are more present, but with a broader distribution. This indicates, that very small values of t, $e_{\max}$ and DW have a stronger impact on the degradation of the overall accuracy, than the impact that large values of these parameters have on the improvement of the accuracy. For this purpose, in the following, we shall focus on the parameter histograms corresponding to the highest errors.

With these observations in mind, we trained a second CNN (CNN2) where the range of parameters have been reduced so as to exclude the critical regions of t, $e_{\max}$ and DW. The parameter range is given in table 1 and the performances of the CNN is given in table 2. The convergence of the training improves significantly (with a training and test mse reduced to 0.00077 and 0.013, respectively). The accuracy on all parameters also increases significantly (up to 17 percentage points), except the width of the Bragg peak which increases only by one percentage point, which is easily explained since this parameter is not directly affected by the parameters modified in CNN2. Most importantly the accuracy on the strain profile increases up to 82%, which means that 82% of the 1.5×10^5 test data yielded predictions with a relative strain rmse smaller than 0.1. The plot of predictions and accuracy and the parameter histograms are given in figures S3 and S4. From the latter, it can be observed that the region with maximum strain values <2% remain problematic since they have the largest detrimental effect on the overall CNN accuracy. It can therefore be useful to consider

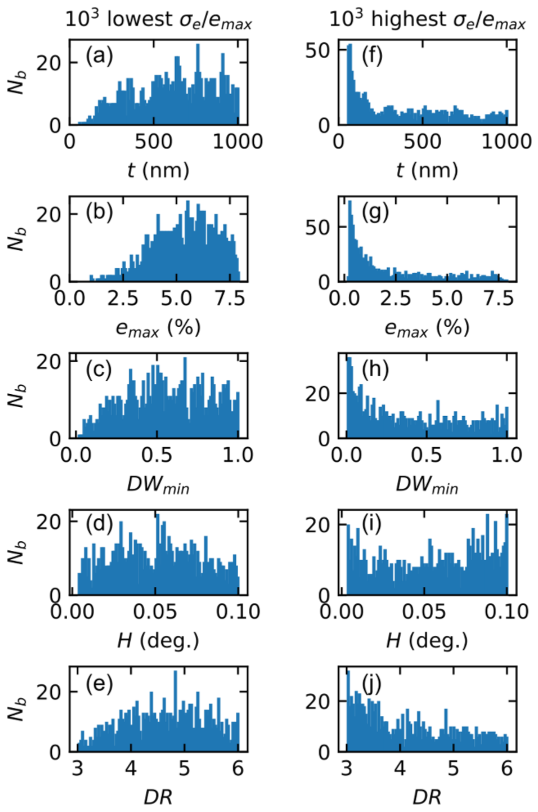

Figure 5. (a)–(e) Histogram of parameters in the 10^3 data with the lowest σ_e/e_{max} values. (f)–(j) Histogram of parameters in the 10^3 data with the highest σ_e/e_{max} values.

specifically this region. Moreover, whereas strain values as high as 7%–8% can be observed, for instance in irradiated SiC crystals [66–68], those are more an exception than a general rule, since most irradiated crystals exhibit strain levels lower than 2%.

3.3. Accuracy and performances in the low strain region ($e_{max} \leqslant 2\%$)

In order to tentatively improve the accuracy and the performances in the domain of low strain levels, we first reduced the data to those data sets with a maximum strain value of 2% and, *without retraining the CNN*, we computed the corresponding accuracies for this data range. The plot of predictions and accuracy, and the parameter histograms for both CNN1 and CNN2 are given in figures S5–S8. The associated performances are given in table 2. It can be observed that the overall performances, although better for CNN2 than for CNN1, are disappointing in this region, with accuracy on the strain profile of only 61%.

In figure 6, we plot the parameter histograms for the highest 10^3 errors, in the case of CNN1 and CNN2.

In the case of CNN1, the distributions of *t*, e_{max} and DW are peaked towards lower values, which is not surprising since this behaviour was already observed in the global performances of the CNN (figure 5). Conversely, in the case of CNN2, it is interesting to notice that the only parameter that inhomogeneously deteriorates the accuracy of the CNN is e_{max}. All others exhibit a rather uniform distribution. This leads us to conclude that, to improve accuracy in this strain range, the CNN should be trained with data specifically restricted to this range (see table 1, CNN3). The performances of this specifically trained CNN are discussed below.

The plot of predictions and accuracy, and the parameter histograms are given in figures S9 and S10. The performances of CNN3 are summarized in table 2. It is remarkable that, despite degraded learning metrics (train and test mse), the accuracy of CNN3 is higher than that of CNN1. This shows that the learning metrics alone do not allow to judge the prediction capabilities of a given CNN, especially if the different CNNs have been trained with different data. The most striking feature is that, compared to CNN1 and CNN2 in the low strain regime, CNN3 reaches an accuracy of 76% which is 26 and 15 percentage points higher than CNN1 and CNN2, respectively.

Unfortunately, although an accuracy of 76% is a valuable achievement, it does not allow for the corresponding CNN to completely replace least-square simulations, as 24% of the data may contain relative errors larger than 0.1. Nonetheless, it can provide a good starting point for simulations. This is illustrated in

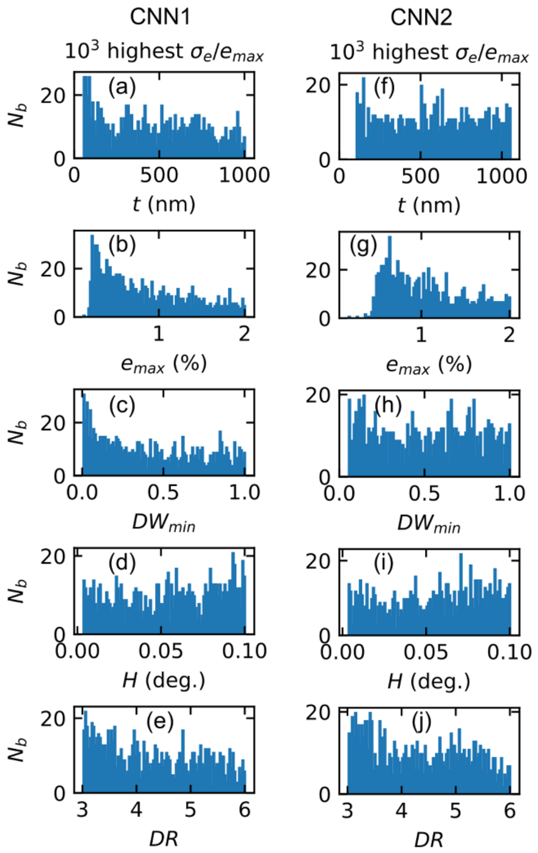

Figure 6. (a)–(e) Histogram of parameters in the 10^3 data with the highest $\sigma_e/e_{\max}$ values in the case of CNN1 (a)–(e) and CNN2 (f)–(j).

the last section, using real experimental data. Before that, we shall investigate the performances and the robustness of the CNN for different architectures, training situations or challenging data. For this purpose we will use the last CNN (CNN3).

3.4. Influence of the output vector size

As explained in section 2.1.2, two parameters related to the experimental set-up (namely the width of the Bragg peak, H, and the dynamic range, DR) were included in the training although they do not characterize the strain profile. This is justified by the fact that the ultimate goal of this work is to provide a fully automated data analysis pipeline, without the need of human intervention. However, it can legitimately be asked how these parameters affect the overall learning capability of the CNN, especially since those experimental parameters could straightforwardly be obtained by a separate analysis: a least square fit of the Bragg peak with a Gaussian function, for instance, for the former, and the evaluation the maximum/minimum intensity ratio, for the latter.

A new CNN returning only sample-related parameters, labelled CNN[12] (where 12 stands for number of output parameters of the CNN), has been trained, and the corresponding performances are given in table 2. The first observation that can be made is that the learning metrics (training and validation mse) are degraded as compared to CNN3. This can be understood by the fact that H and DR are both predicted with excellent accuracies, namely 91% and 100%, respectively. Removing these well-learned parameters from the training mathematically deteriorates the learning metrics. For the same reason, since H and DR are easily learned by the CNN, they are not expected to impair the accuracy of the other parameters inferred by the network. This is indeed what is observed in table 2: the accuracy of the predicted thickness, maximum strain, DW factor and strain profile remain constant, with a 1% decrease for $e_{\max}$ and DW (although this is not expected to be meaningful due to the stochastic nature of the training process).

Another corollary question, that arises from the previous analysis, is how the overall performances of the CNN would evolve when the parameters with the worst accuracies are removed from the training. The parameter characterized by the worst accuracy (76%) is the strain profile. Since this parameter is precisely the one we are interested in the present article, removing it from the training would make our work useless. We can nonetheless consider this situation to check if the behaviour observed above, when removing

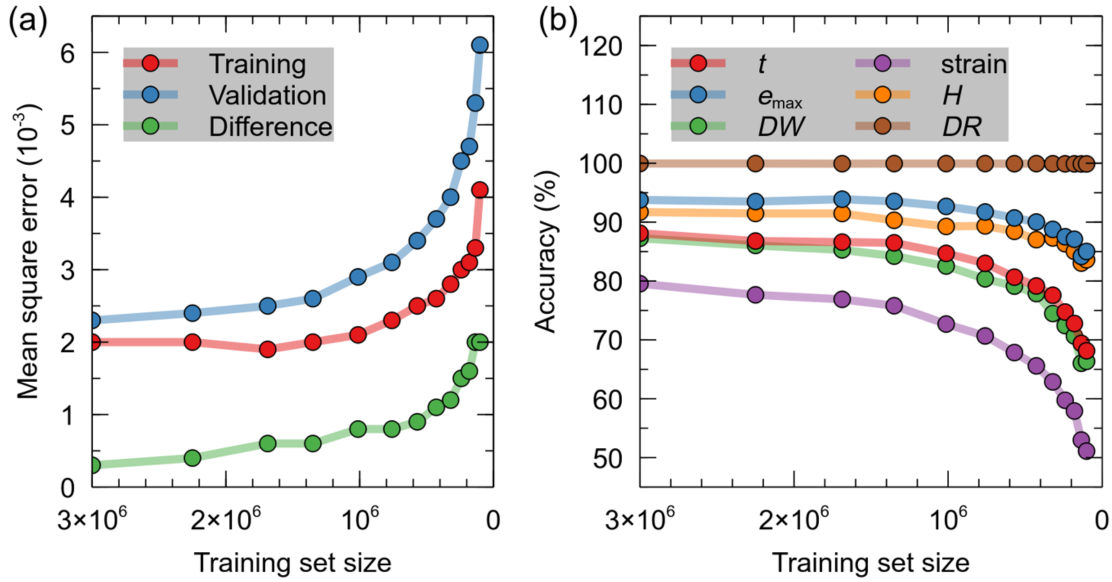

Figure 7. Evolution of the training and validation mean-square errors (a) and of the prediction accuracies (b) as a function of the training dataset size. In (a) and (b) the abscissa is plotted with decreasing values.

well-learned parameters, is confirmed when removing less well-learned parameters. As expected, removing the 10 *B*-spline weights describing the strain profile, results in an automatic improvement in the learning metrics. In addition, a slight improvement of the accuracy of the determination of the *t*, e_{max} and DW is observed, with an increase of 2, 3 and 5 percentage points; the main drawback being that the ability to determine the shape of the strain profile is lost. This could nonetheless suggest that training two separate CNNs, with optimized architectures, one for the 'easy' parameters and another one dedicated to the shape of the strain profile, could be envisioned as a possible evolution to improve the global performances.

3.5. Robustness against the training data set size

Another legitimate question that can be asked, is how much the size of the training dataset can be reduced while still giving good predictions. This is particularly important since it is well known that NN are easily subject to overfitting when the dataset size is too small: the NN memorizes the values of the training dataset and the validation mse increases after reaching a minimum; its performance as a predictor decreases [72]. Although no overfitting has been observed in the training (see figure S1), the question of the maximum accuracy that can be reached on the test data (which, we recall, is not used in the training) for a given CNN architecture, is essential.

For this purpose we generated a training dataset containing 3×10^6 input vectors, and iteratively reduced its size by 25%. For each dataset we trained the CNN and evaluated all accuracies. The results are displayed in figure 7.

Let us first discuss the training metrics, figure 7(a). Upon decreasing the training set size, both the training and the validation mse increase. The mse increase slowly for dataset sizes down to $\sim 8 \times 10^5$; below this value, the mse increase at a much faster rate. This indicates that the CNN becomes harder to train: when the learning curve saturates at a constant value (see figure S1), the values inferred by the CNN exhibit higher errors. Interestingly, the validation mse increases faster than the training mse (see the green difference curve). This observation evidences that, when the dataset size decreases, the predictions performed by the CNN are degraded at higher rate than the degradation of the learning, i.e. the value of the CNN as a predictor decreases. Conversely, increasing the data size narrows the gap between the training and validation mse, indicating an improved prediction reliability.

Let us now discuss the accuracies, figure 7(b). Similarly to what is observed for the training metrics, the accuracies are slowly degrading down to a dataset size threshold value, below which the rate of degradation increases. It also appears that this threshold value depends on the initial accuracy of a given parameter. For instance, e_{max}, whose initial accuracy is 94%, remains roughly constant down to $\sim 1 \times 10^6$, whereas the accuracy of the strain profile, with initial value of 80%, exhibits a higher degradation rate below $\sim 1.3 \times 10^6$. For this reason, in the present study, we selected a training dataset size around this threshold value. Increasing the dataset size would only yield minor improvements in the accuracies of all parameters, except the strain profile which would benefit more from the increased dataset size (80% instead of 76%). Obviously,

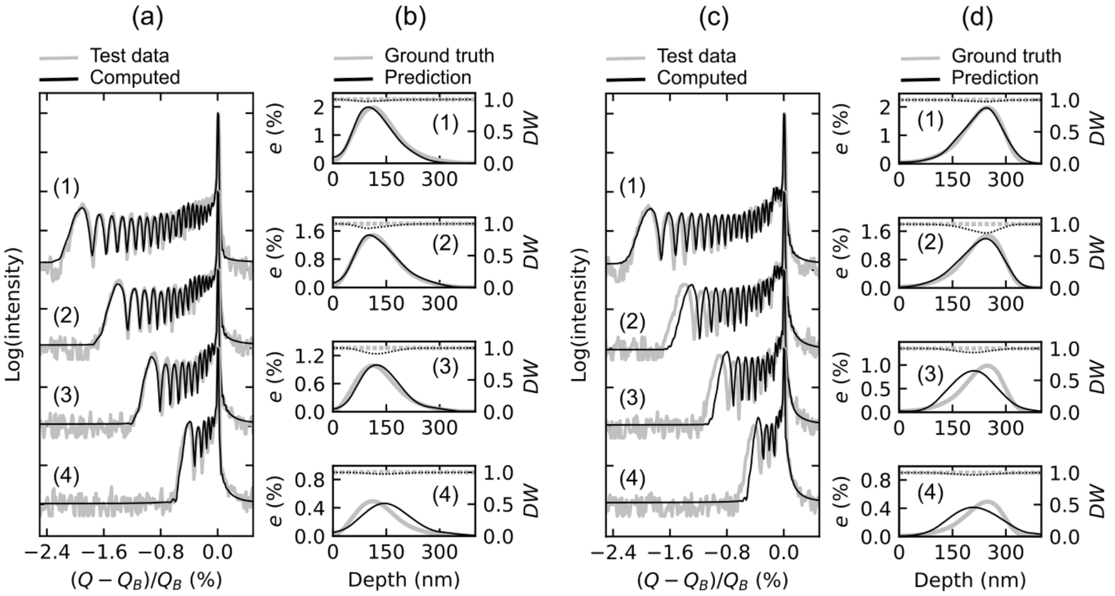

Figure 8. XRD test data (grey curves) and computed from predictions (black curves) in the case of right-hand (a) and left-hand tailed strain profiles (c). The corresponding strain profiles (grey) and DW (dotted grey) profiles are given, as well as the associated predictions (black) are given in (b), (d). The labels (1–4) correspond to decreasing maximum strain.

this increased dataset size result would result in considerably increased computation times: a factor of ∼2.5 in the present case (i.e. more than 6 h to generate the dataset, instead of 2.5 h). Considering the relative weak improvement in the accuracy we decided not to increase the dataset size.

3.6. Challenging cases, uniqueness of the inference and overall limitations

An issue common to all fitting approaches is to ascertain to what extent the solution returned by the algorithm is unique. The same issue holds for predictions performed by a NN. In other words, in the present case, could two different strain profiles yield the same XRD curve? If so, this would jeopardize the whole approach. In the case of XRD, this issue is closely related to a topic frequently discussed in the solid-state community: is XRD really able to determine the absolute position and shape of a strain distribution? In this section we briefly address these issues by considering situations which are expected to be challenging for the CNN (or a least-square fitting algorithm).

In the first example we address the case of two mirrored asymmetric strain profiles, spanning a depth of 400 nm, with a maximum strain decreasing from 2% down to 0.5%, figure 8. Let us first consider the strain profiles exhibiting the highest strain (curves labelled (1)). While it is evident that the XRD curves are very similar, there are also some noticeable difference, in particular in the region with low $\Delta Q/Q_B$ values. The difference stems from the fact that, for the left-hand tailed strain profiles (*d*), the waves scattered from the strain-free substrate (>400 nm) interfere with the waves scattered from the near surface region (<∼50 nm), where the strain is also close to zero, whereas no such interferences exist in the case of the right-hand tailed strain profile (b). These interferences create the specific fringe pattern observed around $\Delta Q/Q_B = -0.1\%$ (c). Interestingly, the CNN is perfectly able to retrieve the correct strain profile, despite the similarity in the XRD curves. The same situation is true for the case of the 1.5% strain (curves labelled (2)). Below 1.5% strain (curves labelled (3) and (4)), the prediction are noticeably degraded and the CNN ends up predicting approximately the same solution in the 0.5% case. The reason for this is that, for decreasing strain, the amount of information contained in the XRD curves progressively decrease (less fringes) and, while the strain profiles are indeed different, at 0.5%, the strain is too low to yield a significantly different XRD curve. In this situation, the CNN fails to predict the correct shape of the strain profile, as would a conventional least-square algorithm. It can be noticed that, overall, the CNN performs better in the case of the right-hand tailed strain profiles (a, b). A likely reason for this is that the CNN does not have to rely on small details of the XRD curves (i.e. in the region around $\Delta Q/Q_B = -0.1\%$) to provide the correct solution.

Decreasing the amount of information in the XRD curve is also possible by decreasing the thickness, decreasing the dynamic range or decreasing the Debye–Waller factor. In turns out that while a similar evolution is observed, it is much less pronounced than the effect of decreasing strain. We therefore focused on the latter.

The second example is the case of a highly disordered region ($DW_{min} = 0.05$) buried a various depths below the surface, figure 9. The difficulty here stems from the fact that the XRD curves almost do not contain

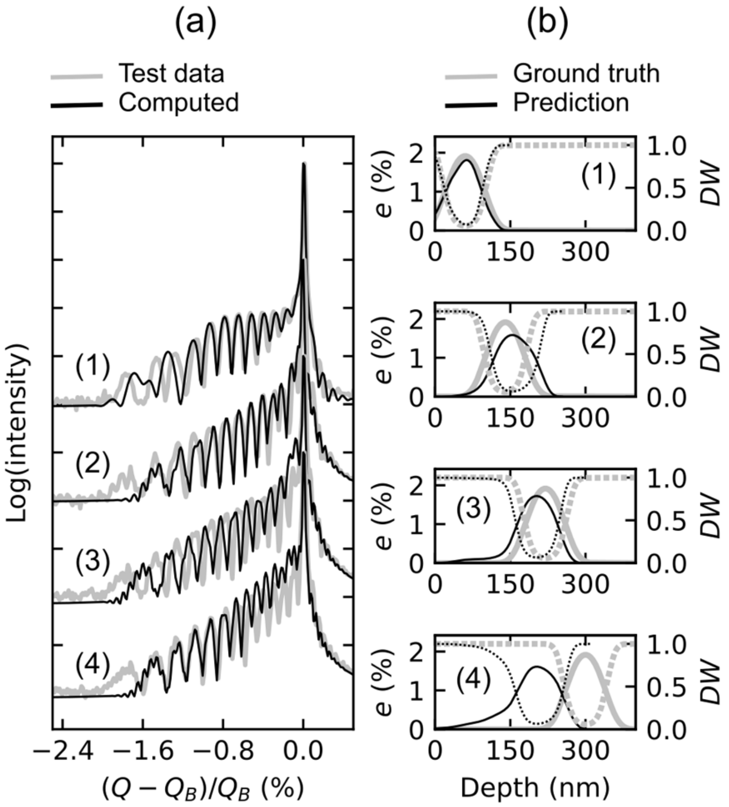

Figure 9. XRD test data (grey curves) and computed from predictions (black curves) in the case of highly disordered regions buried at increasing depth (1–4). The corresponding strain profiles (grey) and DW (dotted grey) profiles, as well as the associated predictions (black) are given in (b).

information about the damaged regions itself (because the low DW factor considerably weakens the intensity scattered from this region), but from the strained regions on either side of the damaged region.

Let us first consider the case of disordered region located just below the surface (1). The CNN predicts both the strain and DW profiles correctly. Interestingly, the CNN predicts a thickness of ∼150 nm, whereas a value of 400 nm has been used to generate the data. This has, however, no consequence on the predicted strain profile, since for depth larger than 150 nm the strain and DW are equal to 0 and 1, respectively. In other words, the CNN predicts that, for depths larger than 150 nm, the materials is strain- and damage-free and it considers this region as being part of the substrate which, strictly speaking, is actually the case. The prediction, in terms of the depth of the buried damaged region, remain perfectly acceptable in cases (2) and (3). For the deepest buried layer (4), the CNN fails to predict correct values. The explanation is similar to the previous case: the differences in the XRD curves are based on interferences between the substrate and the top most region of the strain profile, yielding narrow interference fringes which, in the present are difficult to distinguish from noise. As a consequence, the CNN fails to detect these tiny features.

To conclude this part, we can mention that, while it is not possible to give a general answer to the question regarding the uniqueness of the strain profile predicted by the CNN, a rule of thumb is that it can only be reliably determined if there is enough information regarding the said strain profile in the XRD curve. For this reason, low strains (∼0.5%) or low DW factors (0.05) appear to be limiting cases, as was already inferred in section 3.1. Similar conclusion holds for very low thickness and/or dynamic range. In these situations, an *a posteriori* comparison with complementary information, obtained using either experimental techniques such as Rutherford backscattering spectroscopy in channelling mode or Raman spectroscopy [81], or computer codes such as Stopping and Range of Ions in Matter (SRIM) [82], remains inevitable. Another important point that should be mentioned is that the CNN is only able to predict strain profile shapes similar to those used in the training procedure. In the present case this excludes, for instance, bimodal strain profiles or perfectly flat strain profiles. This is further discussed in the next section.

3.7. Test on real data

Experimental XRD data corresponding to the 400 reflection of disordered Y_2O_3-doped ZrO_2 single crystals irradiated with 300 keV Cs ions are shown in figure 10. The colours blue, orange and green are associated to

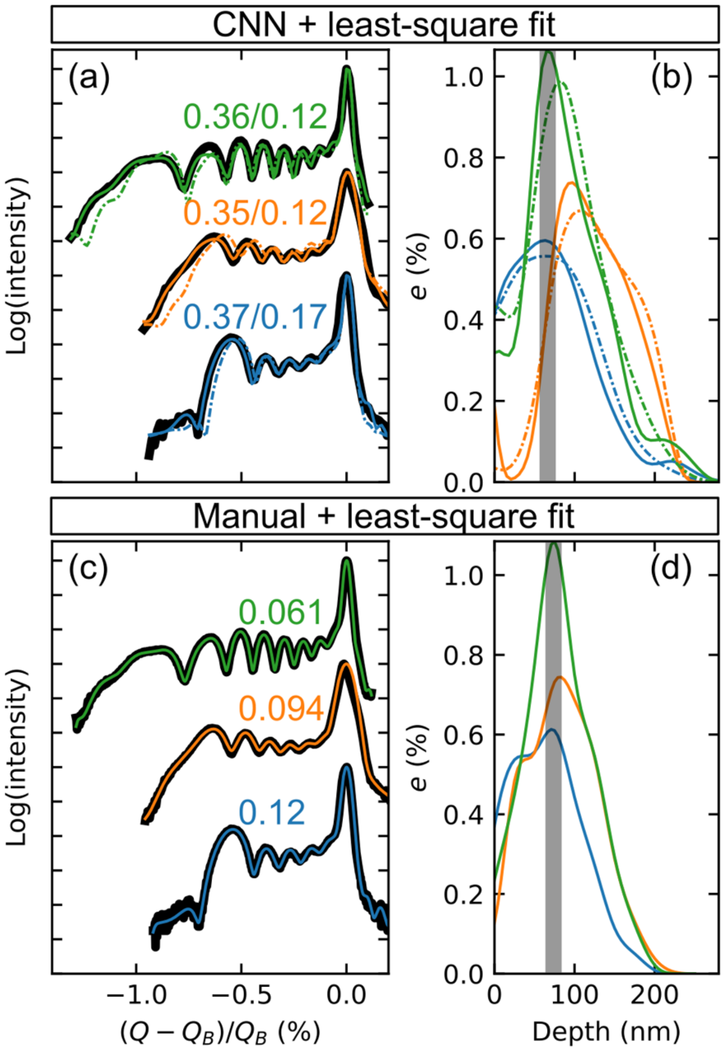

Figure 10. (a) XRD data from the 400 reflection of ZrO_2 exhibiting increasing level of irradiation-induced disorder (blue, orange, green). The dash-dotted lines correspond the intensity computed from the CNN's prediction. The continuous lines correspond to a least-square fit using the CNN predictions as a starting point for the fit. (b) Strain profiles predicted by the CNN (dash-dotted lines) and after least-squares fitting (continuous lines). (c) Least-square fit of the XRD data using a manual adjustment of the initial solution. (d) Corresponding strain profiles. In (b), (d) the grey line is 20 nm wide and marks the depth position of the strain maximum.

increasing ion fluences, 7.5 × 10^{13}, 3 × 10^{14} and 6 × 10^{14} cm^{-2}, respectively, which correspond to increasing levels of strain and disorder within the crystals.

In the case of experimental data, no ground truth is available. Therefore, one possible way to assess the quality of the predictions of the CNN is to compute the XRD curves, $I^{(p)}$, using the predictions of the CNN as an input, and comparing them with experimental XRD data, I^{exp}. The result is shown in figure 10(a) as dash-dotted lines, together with the predicted strain profiles given in figure 10(b), also as dash-dotted lines. The first observation that can be made is that, while they are not perfect, the predictions of CNN are remarkably close to the experimental XRD data. The prediction time varies from 100 ms/prediction down 22 μs/prediction to for a data set size increasing from 1 to 150 000[7]. The quality of the prediction can be provided by the intensity rmse, $\sigma_I = \sqrt{\left\langle \left[\log\left(I^{(p)}\right) - \log\left(I^{\mathrm{exp}}\right)\right]^2 \right\rangle}$, which ranges between 0.35 and 0.37.

Using the predictions of the CNN (figure 10(b)) as an input, it is possible to further optimize the strain profile using a conventional least-square fitting of the XRD intensity with adjustable strain profile parameters. This has been implemented in an automated data prediction and least-square fitting pipeline (CNN + leastsq). The result is shown as full lines in figures 10(a) and (b). The optimized predictions are

[7] This counter-intuitive result is explained by the fact that analysing a large data set makes an optimal use of the parallel GPU architecture, thereby compensating the time loss induced by the data transfer from the central memory to the GPU.

close to perfect, with a rms intensity dropping to 0.12–0.17, while the total prediction time increases to 5.3, 54 and 9 s, for the three increasing fluences, respectively.

It can be interesting to compare these results with those obtained using the classical approach, namely a manual determination of the starting strain profile, followed by a least-square fitting procedure (manual + leastsq). In this case, the simulations require several minutes, up to several tens of minutes, with most of the time being spent in the manual adjustment of the starting strain profile, prior to the least-square fit. The results are displayed in figures 10(c) and (d). While the quality of the simulation is clearly better, as attested by a visual inspection of the XRD curves as well as by the rms intensity values ranging between 0.061 and 0.12, the corresponding strain profile are strikingly similar to those obtained by CNN + leastsq: the magnitude of the strain, the depth location of the strain maximum (indicated by the grey band with 20 nm width) and the overall shape of the curve are very well reproduced by the CNN, with two exceptions:

- The intermediate curve (orange) predicted by CNN + leastsq significantly differs from the manual + leastsq solution. This is due to the fact that the initial prediction of the CNN is incorrect, so that the least-square fit struggles to find a minimum (as indicated by the longer fitting time reported above) and ends up in a secondary minimum. This is a universal drawback of least-square fitting inversion methods, where the quality of the solution depends on the quality of the *a priori* knowledge.
- The profiles obtained by CNN + leastsq exhibit bump in the strain at larger depth which do not exist in the manual + leastsq solution. This observation can be traced back to the discrepancy that exist between the experimental and predicted intensity curve, in the very close vicinity of the intense peak at $\Delta Q/Q_{\mathrm{B}} = 0$, which arises from a slight underestimation of the width of the peak by the CNN.

To conclude this section, it is worth reminding that the training data has not been created with the objective of operating on experimental data, but rather to provide a proof of concept that a CNN can be used to extract strain profiles from XRD data. As such, several features that characterize actual XRD data have been voluntarily neglected:

- First and foremost, the DW profile was assumed to be controlled by the strain profile, according to equation (4). However, it is experimentally well established that the DW profile might significantly deviate from its shape given by equation (4); see for instance [11].
- Second, the only source of noise that has been included is Poisson noise inherent to photon counting. However, other sources of intensity deviations can occur such as imperfect detector behaviour, imperfect stepper motors, parasitic scattering from optical elements in the beam path, *etc.* To account for these errors, it has been suggested to add uniform noise to the data [58].
- Third, the background intensity has been assumed to be constant, whereas non-constant backgrounds can be observed in practice.
- Fourth, only mono-modal and peaked strain profiles have been generated for the training, whereas actual strain profiles can sometimes exhibit bimodal or perfectly flat strain profiles.

That being said, the CNN-deduced strain profiles do not drastically differ from those obtained by human operator using a least-square fitting algorithm. This result constitutes evidence of a very promising new route to address the issue of strain depth profile retrieval.

4. Conclusions and future work

We have provided a proof of concept that CNNs can be used to determine strain profiles from XRD data in ion irradiated or ion implanted materials. The CNN is built on usual convolutional, Max Pooling and batch normalization layers, together with dense neuron layers, implemented in *TensorFlow/Keras*. The CNN has been trained with 1.215×10^6 numerically generated data (plus 1.35×10^5 used a validation dataset), corresponding to an equivalent number of strain profiles, described by a reduced set of 11 parameters (10 *B*-spline weights for the overall shape and magnitude of the strain profile plus the disordered thickness) and three additional parameters (the minimum DW factor, the dynamic range and the width of the Bragg peak). The performances of the CNN have been tested using 1.5×10^5 data that were not included in the training.

Prediction accuracies ranged from 90% to 100% for individual parameters such as the maximum strain, the thickness, the minimum Debye–Waller factor, the width of the Bragg peak and the dynamic range. Regarding the entire strain profile, the best accuracy reached 82% for a maximum strain range of [0.5%, 8%] and 76% (using a specifically trained CNN) in the [0.5%, 2%] range, where the amount of information is significantly reduced.

The results obtained so far are promising and suggest that CNNs could, in the future, be used to lower or even remove the need of *a priori* information for a subsequent, automated, data simulation procedure; this information being provided by the CNN. However, an *a posteriori* comparison with complementary information is still required to check to reliability of the simulations. Moreover, in order for a CNN to potentially completely replace simulations (and the need for a human supervision), several improvements should be brought to the network and, more specifically, to the training data:

- Decouple the disorder (DW) profile from the strain profile. While this is easy to envision for the training data, it might require adjustments in the CNN architecture, since the amount of parameters would significantly increase (up to 23 in the present case, instead of 14).
- Include non-Poisson noise and a non-constant background.
- We here only considered one lattice plane family of one single material. While it is possible train specific networks for different materials and different lattice planes, a smarter approach would be to develop a suitable scaling scheme in order to bring all possible datasets on a common angular or *Q* range.
- Finally, a wider range of strain profile shapes should be considered (bimodal shapes, flat profiles, *etc.*), in order to expand the prediction ability of the CNN.

These topics are under active consideration and, when finalized, while be implemented in a completely automated data analysis pipeline.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://github.com/aboulle/MLST-XRD.

Acknowledgments

We are grateful to the Mission for Transversal and Interdisciplinary Initiatives (MITI) of the French National Center for Scientific Research (CNRS) program (CoMITIS project) and the Nucléaire, énergie, environnement, déchets, société (NEEDS) program (RAISIN project) for funding this work.

ORCID iD

A Boulle https://orcid.org/0000-0003-1042-6403

References

[1] Warren B E 1969 *X-ray Diffraction* (Reading, MA: Addison-Wesley)

[2] Krivoglaz M A 1996 *Theory of X-ray and Thermal Neutron Scattering by Real Crystals* (Berlin: Springer)

[3] Snyder R, Fiala J and Bunge H J 2000 *Defect and Microstructure Analysis by Diffraction* (Oxford: Oxford University Press)

[4] Pietsch U, Holý V and Baumbach T 2004 *High-Resolution X-ray Scattering—From Thin Films to Lateral Nanostructures* (New York: Springer)

[5] Elliman R G and Williams J S 2015 Advances in ion beam modification of semiconductors *Curr. Opin. Solid State Mater. Sci.* **19** 49–67

[6] Bowen D K and Tanner B K 2017 *X-ray Metrology in Semiconductor Manufacturing* (Boca Raton: CRC Press)

[7] Speriosu V S 1981 Kinematical x-ray diffraction in nonuniform crystalline films: strain and damage distributions in ion-implanted garnets *J. Appl. Phys.* **52** 6094–103

[8] Zaumseil P, Winter U, Cembali F, Servidori M and Sourer Z 1987 Determination of dislocation loop size and density in ion implanted and annealed silicon by simulation of triple crystal x-ray rocking curves *Phys. Status Solidi* a **100** 95–104

[9] Klappe J G E and Fewster P F 1994 Fitting of rocking curves from ion-implanted semiconductors *J. Appl. Cryst.* **27** 103–10

[10] Milita S and Servidori M 1995 X-ray rocking-curve analysis of crystals with buried amorphous layers. Case of ion-implanted silicon *J. Appl. Crystallogr.* **28** 666–72

[11] Boulle A and Debelle A 2010 Strain-profile determination in ion-implanted single crystals using generalized simulated annealing *J. Appl. Cryst.* **43** 1046

[12] Rieutord F, Mazen F, Reboh S, Penot J D, Bilteanu L, Crocombette J P, Vales V, Holy V and Capello L 2013 Lattice strain of hydrogen-implanted silicon: correlation between x-ray scattering analysis and *ab-initio* simulations *J. Appl. Phys.* **113** 153511

[13] Souilah M, Boulle A and Debelle A 2016 RaDMaX: a graphical program for the determination of strain and damage profiles in irradiated crystals *J. Appl. Cryst.* **49** 311

[14] Boulle A and Mergnac V 2020 RaDMaX online : a web-based program for the determination of strain and damage profiles in irradiated crystals using x-ray diffraction *J. Appl. Crystallogr.* **53** 587–93

[15] Kirkpatrick S, Gelatt C D and Vecchi M P 1983 Optimization by simulated annealing *Science* **220** 671–80

[16] Sun S *et al* 2019 Accelerated development of perovskite-inspired materials via high-throughput synthesis and machine-learning diagnosis *Joule* **3** 1437–51

[17] Garcia-Cardona C, Kannan R, Johnston T, Proffen T, Page K and Seal S K 2019 Learning to predict material structure from neutron scattering data *2019 IEEE Int. Conf. on Big Data (Big Data)* pp 4490–7

[18] Purushottam Raj Purohit R R P, Tardif S, Castelnau O, Eymery J, Guinebretière R, Robach O, Ors T and J-S M 2022 LaueNN: neural-network-based hkl recognition of Laue spots and its application to polycrystalline materials *J. Appl. Crystallogr.* **55** 737–50

[19] Park W B, Chung J, Jung J, Sohn K, Singh S P, Pyo M, Shin N and Sohn K-S 2017 Classification of crystal structure using a convolutional neural network *IUCrJ* **4** 486–94

[20] Vecsei P M, Choo K, Chang J and Neupert T 2019 Neural network based classification of crystal symmetries from x-ray diffraction patterns *Phys. Rev.* B **99** 245120

[21] Liu C-H, Tao Y, Hsu D, Du Q and Billinge S J L 2019 Using a machine learning approach to determine the space group of a structure from the atomic pair distribution function *Acta Crystallogr.* A **75** 633–43

[22] Oviedo F *et al* 2019 Fast and interpretable classification of small x-ray diffraction datasets using data augmentation and deep neural networks *npj Comput. Mater.* **5** 60

[23] Banko L, Maffettone P M, Naujoks D, Olds D and Ludwig A 2021 Deep learning for visualization and novelty detection in large x-ray diffraction datasets *npj Comput. Mater.* **7** 104

[24] Lee J-W, Park W B, Lee J H, Singh S P and Sohn K-S 2020 A deep-learning technique for phase identification in multiphase inorganic compounds using synthetic XRD powder patterns *Nat. Commun.* **11** 86

[25] Wang H, Xie Y, Li D, Deng H, Zhao Y, Xin M and Lin J 2020 Rapid identification of x-ray diffraction patterns based on very limited data by interpretable convolutional neural networks *J. Chem. Inf. Model.* **60** 2004–11

[26] Rabiej M and Rabiej S 2021 Application of the artificial neural network for identification of polymers based on their x-ray diffraction curves *Comput. Mater. Sci.* **186** 110042

[27] Schuetzke J, Benedix A, Mikut R and Reischl M 2021 Enhancing deep-learning training for phase identification in powder x-ray diffractograms *IUCrJ* **8** 408–20

[28] Dong H *et al* 2021 A deep convolutional neural network for real-time full profile analysis of big powder diffraction data *npj Comput. Mater.* **7** 74

[29] Chitturi S R, Ratner D, Walroth R C, Thampy V, Reed E J, Dunne M, Tassone C J and Stone K H 2021 Automated prediction of lattice parameters from x-ray powder diffraction patterns *J. Appl. Crystallogr.* **54** 1799–810

[30] Zhuang Y *et al* 2022 Unsupervised learning approaches to characterizing heterogeneous samples using x-ray single-particle imaging *IUCrJ* **9** 204–14

[31] Ignatenko A, Assalauova D, Bobkov S A, Gelisio L, Teslyuk A B, Ilyin V A and Vartanyants I A 2021 Classification of diffraction patterns in single particle imaging experiments performed at x-ray free-electron lasers using a convolutional neural network *Mach. Learn.: Sci. Technol.* **2** 025014

[32] Meister N, Guan Z, Wang J, Lashley R, Liu J, Lhermitte J, Yager K, Qin H, Sun B and Yu D 2017 Robust and scalable deep learning for x-ray synchrotron image analysis *2017 New York Scientific Data Summit (NYSDS)* pp 1–6

[33] Song G, Porcar L, Boehm M, Cecillon F, Dewhurst C, Le Goc Y, Locatelli J, Mutti P and Weber T 2020 Deep learning methods on neutron scattering data *EPJ Web Conf.* **225** 01004

[34] Guan Z, Qin H, Yager K G, Choo Y and Yu D 2018 Automatic x-ray scattering image annotation via double-view fourier-bessel convolutional networks *British Machine Vision Conf. 2018, BMVC 2018* (*Newcastle, UK,3–6 September 2018*) (BMVA Press) p 245

[35] Zimmermann J *et al* 2019 Deep neural networks for classifying complex features in diffraction images *Phys. Rev.* E **99** 063309

[36] Hadi Kiapour M, Yager K, Berg A C and Berg T L 2014 Materials discovery: fine-grained classification of x-ray scattering images *2014 IEEE Winter Conf. on Applications of Computer Vision (WACV)* (Steamboat Springs, CO: IEEE) pp 933–40

[37] Ikemoto H, Yamamoto K, Touyama H, Yamashita D, Nakamura M and Okuda H 2020 Classification of grazing-incidence small-angle x-ray scattering patterns by convolutional neural network *J. Synchrotron Radiat.* **27** 1069–73

[38] Wang B, Guan Z, Yao S, Qin H, Nguyen M H, Yager K and Yu D 2016 Deep learning for analysing synchrotron data streams *2016 New York Scientific Data Summit (NYSDS)* (New York: IEEE) pp 1–5

[39] Wang B, Yager K, Yu D and Hoai M 2017 X-ray scattering image classification using deep learning *2017 IEEE Winter Conf. on Applications of Computer Vision (WACV)* pp 697–704

[40] Liu S, Melton C N, Venkatakrishnan S, Pandolfi R J, Freychet G, Kumar D, Tang H, Hexemer A and Ushizima D M 2019 Convolutional neural networks for grazing incidence x-ray scattering patterns: thin film structure identification *MRS Commun.* **9** 586–92

[41] He H, Liu C and Liu H 2020 Model reconstruction from small-angle x-ray scattering data using deep learning methods *iScience* **23** 100906

[42] Chang M-C, Wei Y, Chen W-R and Do C 2020 Deep learning-based super-resolution for small-angle neutron scattering data: attempt to accelerate experimental workflow *MRS Commun.* **10** 11–17

[43] Samarakoon A M *et al* 2020 Machine-learning-assisted insight into spin ice Dy2Ti2O7 *Nat. Commun.* **11** 892

[44] Bellisario A, Maia F R N C and Ekeberg T 2022 Noise reduction and mask removal neural network for x-ray single-particle imaging *J. Appl. Crystallogr.* **55** 122–32

[45] Song Y, Tamura N, Zhang C, Karami M and Chen X 2019 Data-driven approach for synchrotron x-ray Laue microdiffraction scan analysis *Acta Cryst.* A **75** 876–88

[46] Sullivan B, Archibald R, Vandavasi V, Langan P, Coates L and Lynch V 2019 Volumetric segmentation via neural networks improves neutron crystallography data analysis *2019 19th IEEE/ACM Int. Symp. on Cluster, Cloud and Grid Computing (CCGRID)* (Larnaca: IEEE) pp 549–55

[47] Sullivan B, Archibald R, Azadmanesh J, Vandavasi V G, Langan P S, Coates L, Lynch V and Langan P 2019 BraggNet: integrating Bragg peaks using neural networks *J. Appl. Crystallogr.* **52** 854–63

[48] Starostin V, Munteanu V, Greco A, Kneschaurek E, Pleli A, Bertram F, Gerlach A, Hinderhofer A and Schreiber F 2022 Tracking perovskite crystallization via deep learning-based feature detection on 2D x-ray scattering data *npj Comput. Mater.* **8** 1–9

[49] Liu Z, Sharma H, Park J-S, Kenesei P, Miceli A, Almer J, Kettimuthu R and Foster I 2022 BraggNN: fast x-ray Bragg peak analysis using deep learning *IUCrJ* **9** 104–13

[50] Cherukara M J, Nashed Y S G and Harder R J 2018 Real-time coherent diffraction inversion using deep generative networks *Sci. Rep.* **8** 1–8

[51] Cherukara M J, Zhou T, Nashed Y, Enfedaque P, Hexemer A, Harder R J and Holt M V 2020 AI-enabled high-resolution scanning coherent diffraction imaging *Appl. Phys. Lett.* **117** 044103

[52] Scheinker A and Pokharel R 2020 Adaptive 3D convolutional neural network-based reconstruction method for 3D coherent diffraction imaging *J. Appl. Phys.* **128** 184901

[53] Chan H, Nashed Y S G, Kandel S, Hruszkewycz S O, Sankaranarayanan S K R S, Harder R J and Cherukara M J 2021 Rapid 3D nanoscale coherent imaging via physics-aware deep learning *Appl. Phys. Rev.* **8** 021407

[54] Lim B *et al* 2021 A convolutional neural network for defect classification in Bragg coherent x-ray diffraction *npj Comput. Mater.* **7** 1–8

[55] Wu L, Yoo S, Suzana A F, Assefa T A, Diao J, Harder R J, Cha W and Robinson I K 2021 Three-dimensional coherent x-ray diffraction imaging via deep convolutional neural networks *npj Comput. Mater.* **7** 175

[56] Greco A, Starostin V, Karapanagiotis C, Hinderhofer A, Gerlach A, Pithan L, Liehr S, Schreiber F and Kowarik S 2019 Fast fitting of reflectivity data of growing thin films using neural networks *J. Appl. Crystallogr.* **52** 1342–7

[57] Doucet M, Archibald R K and Heller W T 2021 Machine learning for neutron reflectometry data analysis of two-layer thin films *Mach. Learn.: Sci. Technol.* **2** 035001

[58] Greco A, Starostin V, Hinderhofer A, Gerlach A, Skoda M W A, Kowarik S and Schreiber F 2021 Neural network analysis of neutron and x-ray reflectivity data: pathological cases, performance and perspectives *Mach. Learn.: Sci. Technol.* **2** 045003

[59] Greco A *et al* 2022 Neural network analysis of neutron and x-ray reflectivity data: automated analysis using mlreflect, experimental errors and feature engineering *J. Appl. Crystallogr.* **55** 362–9

[60] Mironov D, Durant J H, Mackenzie R and Cooper J F K 2021 Towards automated analysis for neutron reflectivity *Mach. Learn.: Sci. Technol.* **2** 035006

[61] Carmona Loaiza J M and Raza Z 2021 Towards reflectivity profile inversion through artificial neural networks *Mach. Learn.: Sci. Technol.* **2** 025034

[62] Bartels W J, Hornstra J and Lobeek D J W 1986 x-ray diffraction of multilayers and superlattices *Acta Cryst.* A **42** 539–45

[63] Takagi S 1962 Dynamical theory of diffraction applicable to crystals with any kind of small distortion *Acta Crystallogr.* **15** 1311–2

[64] Taupin D 1964 Théorie dynamique de la diffraction des rayons X par les cristaux déformés *Bull. Soc. Franç. Minér. Crist.* **87** 469

[65] Boulle A, Masson O, Guinebretière R and Dauger A 2003 A new method for the determination of strain profiles in epitaxic thin films using x-ray diffraction *J. Appl. Cryst.* **36** 1424–31

[66] Leclerc S, Declémy A, Beaufort M F, Tromas C and Barbot J F 2005 Swelling of SiC under helium implantation *J. Appl. Phys.* **98** 113506

[67] Leclerc S, Beaufort M F, Barbot J F and Declémy A 2012 Strain and amorphization under light-ion implantation in SiC *Europhys. Lett.* **98** 46001

[68] Boulle A and Debelle A 2016 Statistical nature of atomic disorder in irradiated crystals *Phys. Rev. Lett.* **116** 245501

[69] Sousbie N, Capello L, Eymery J, Rieutord F and Lagahe C 2006 x-ray scattering study of hydrogen implantation in silicon *J. Appl. Phys.* **99** 103509

[70] LeCun Y, Boser B, Denker J S, Henderson D, Howard R E, Hubbard W and Jackel L D 1989 Backpropagation applied to handwritten zip code recognition *Neural Comput.* **1** 541–51

[71] Abadi M *et al* 2016 TensorFlow: large-scale machine learning on heterogeneous distributed systems (arXiv:1603.04467)

[72] Chollet F 2018 *Deep Learning with Python* (Shelter Island, NY: Manning Publications)

[73] Goodfellow I, Bengio Y and Courville A 2016 *Deep Learning* (Cambridge, MA: MIT Press)

[74] LeCun Y, Bengio Y and Hinton G 2015 Deep learning *Nature* **521** 436–44

[75] Krizhevsky A, Sutskever I and Hinton G E 2017 ImageNet classification with deep convolutional neural networks *Commun. ACM* **60** 84–90

[76] Simonyan K and Zisserman A 2015 Very deep convolutional networks for large-scale image recognition (arXiv:1409.1556 [cs])

[77] Ioffe S and Szegedy C 2015 Batch normalization: accelerating deep network training by reducing internal covariate shift (arXiv)

[78] Rumelhart D E, Hinton G E and Williams R J 1986 Learning representations by back-propagating errors *Nature* **323** 533–6

[79] Ruder S 2017 An overview of gradient descent optimization algorithms (arXiv:1609.04747 [cs])

[80] Kingma D P and Ba J 2017 Adam: a method for stochastic optimization (arXiv:1412.6980 [cs.LG])

[81] Zhang Y, Debelle A, Boulle A, Kluth P and Tuomisto F 2015 Advanced techniques for characterization of ion beam modified materials *Curr. Opin. Solid State Mater. Sci.* **19** 19

[82] Ziegler J F, Ziegler M D and Biersack J P 2010 SRIM—the stopping and range of ions in matter *Nucl. Instrum. Methods* B **268** 1818

[83] Williams G J, Pfeifer M A, Vartanyants I A and Robinson I K 2003 Three-dimensional imaging of microstructure in au nanocrystals *Phys. Rev. Lett.* **90** 175501