

 Open access • Journal Article • DOI:10.1103/PHYSREVB.99.245120

## Neural network based classification of crystal symmetries from x-ray diffraction patterns — [Source link](#)

Pascal Marc Vecsei, Kenny Choo, Johan Chang, Titus Neupert

**Institutions:** University of Zurich

**Published on:** 11 Jun 2019 - Physical Review B (American Physical Society)

**Topics:** Artificial neural network

Related papers:

- [Classification of crystal structure using a convolutional neural network.](#)
- [Fast and interpretable classification of small X-ray diffraction datasets using data augmentation and deep neural networks](#)
- [Insightful classification of crystal structures using deep learning.](#)
- [Dropout: a simple way to prevent neural networks from overfitting](#)
- [Rapid Identification of X-ray Diffraction Patterns Based on Very Limited Data by Interpretable Convolutional Neural Networks.](#)

Share this paper:    

View more about this paper here: <https://typeset.io/papers/neural-network-based-classification-of-crystal-symmetries-13k3fhkijk>



**University of  
Zurich**<sup>UZH</sup>

**Zurich Open Repository and  
Archive**

University of Zurich  
University Library  
Strickhofstrasse 39  
CH-8057 Zurich  
[www.zora.uzh.ch](http://www.zora.uzh.ch)

---

Year: 2019

---

## **Neural network based classification of crystal symmetries from x-ray diffraction patterns**

Vecsei, Pascal Marc ; Choo, Kenny ; Chang, Johan ; Neupert, Titus

**Abstract:** Machine learning algorithms based on artificial neural networks have proven very useful for a variety of classification problems. Here we apply them to a well-known problem in crystallography, namely the classification of x-ray diffraction (XRD) patterns of inorganic powder specimens by the respective crystal system and space group. Over 105 theoretically computed powder XRD patterns were obtained from inorganic crystal structure databases and used to train a deep dense neural network. For space group classification, we obtain an accuracy of around 54% on experimental data. Finally, we introduce a scheme where the network has the option to refuse the classification of XRD patterns that would be classified with a large uncertainty. This enhances the accuracy on experimental data to 82% at the expense of having half of the experimental data unclassified. With further improvements of neural network architecture and experimental data availability, machine learning constitutes a promising complement to classical structure determination methodology.

DOI: <https://doi.org/10.1103/physrevb.99.245120>

Posted at the Zurich Open Repository and Archive, University of Zurich

ZORA URL: <https://doi.org/10.5167/uzh-172136>

Journal Article

Published Version

Originally published at:

Vecsei, Pascal Marc; Choo, Kenny; Chang, Johan; Neupert, Titus (2019). Neural network based classification of crystal symmetries from x-ray diffraction patterns. *Physical review. B*, 99(24):245120.

DOI: <https://doi.org/10.1103/physrevb.99.245120>

# Neural network based classification of crystal symmetries from x-ray diffraction patterns

Pascal Marc Vecsei, Kenny Choo, Johan Chang, and Titus Neupert

*Department of Physics, University of Zurich, Winterthurerstrasse 190, CH-8057 Zurich, Switzerland*



(Received 28 December 2018; revised manuscript received 6 May 2019; published 11 June 2019)

Machine learning algorithms based on artificial neural networks have proven very useful for a variety of classification problems. Here we apply them to a well-known problem in crystallography, namely the classification of x-ray diffraction (XRD) patterns of inorganic powder specimens by the respective crystal system and space group. Over  $10^5$  theoretically computed powder XRD patterns were obtained from inorganic crystal structure databases and used to train a deep dense neural network. For space group classification, we obtain an accuracy of around 54% on experimental data. Finally, we introduce a scheme where the network has the option to refuse the classification of XRD patterns that would be classified with a large uncertainty. This enhances the accuracy on experimental data to 82% at the expense of having half of the experimental data unclassified. With further improvements of neural network architecture and experimental data availability, machine learning constitutes a promising complement to classical structure determination methodology.

DOI: [10.1103/PhysRevB.99.245120](https://doi.org/10.1103/PhysRevB.99.245120)

## I. INTRODUCTION

Elastic scattering of plane waves is a powerful technique to reveal the structure of translationally invariant systems. Diffraction has been successfully applied to atomic [1], magnetic [2], superconducting vortex [3], and even protein lattices [4] to uncover their crystal structures. Such structure determination can have far reaching implications and prominent examples are as follows: (i) The correct helical DNA structure was first inferred from interpretation of diffraction patterns [5,6]. (ii) For the production of widely used carbon steel avoiding certain crystal structures by thermal quenching is imperative for high-quality material properties [7]. (iii) In drug design, diffraction determination of protein structures is crucial [8]. The determination of crystal structures based on diffraction experiments is therefore an important activity. A subproblem is the determination of crystal symmetries. As there is a finite number of space groups, interpretation of diffraction patterns is then in essence a classification problem. A prevalent strategy is to model the diffracted Bragg reflections through the form and structure factors [1]. Crystal structure determination is then settled by the best fit to the experimental data. The recent broad dissemination of machine learning algorithms in science, game theory, and technology is driven, for a large part, by the ability of neural networks to—after training—classify data [9–25]. There, neural networks are able to acquire implicit knowledge through which classification can be achieved without prior information about which features in the data are relevant for the classification. It is thus of great interest to apply these algorithms to the interpretation of diffraction patterns [23].

In this work, we study the problem of space group determination from powder x-ray diffraction (XRD) patterns using artificial neural networks. We restrict ourselves to inorganic nonmagnetic materials and employ a fully supervised learning scheme for this classification task. To train the network, we generate a large amount of data theoretically, by computing

diffraction patterns based on crystal structure information of real crystals obtained from various databases [26–32]. The trained network is then tested on both theoretical and experimental data, the latter being obtained from the RRUFF database [33]. A recent study [23] reported, using a convolutional neural network architecture, a remarkable classification accuracy of over 80% on theoretical data despite failing to correctly classify the few experimental data they obtained. We contrast the performance of this convolutional network with a simple dense network and demonstrate that the dense network performs significantly better on experimental data. Furthermore, it is found that when the network misclassifies a structure, the wrongly predicted space group often differs from the correct one by only a few symmetry elements. Finally, we show that a classification accuracy of above 80% can be achieved for experimental data, if about half of the data is left unclassified because the network is not certain about it. This is practically relevant since it is better to know that a network is uncertain than for the network to give a wrong classification.

Our results demonstrate that neural networks can be useful in classifying experimental XRD patterns by the space group of the crystal even when training is done only with theoretically computed data. While the accuracy achieved in this classification is not sufficient to rely on this method alone, it may be used to enhance and complement existing algorithms. In any case, the classification accuracy of 50% we obtain for experimental data is already a considerable achievement given that (i) there are 230 space groups to differentiate, (ii) the experimental data is not perfect due to finite counting statistics and backgrounds, crystal defects, and minority phases, and (iii) no prior knowledge about the task is used in training the network and the data is not pre-processed.

The paper is structured as follows: In Sec. II we explain in more detail the problem that the neural network is set out to solve. Section III introduces the structure of the neural networks that we use as well as the training algorithm.

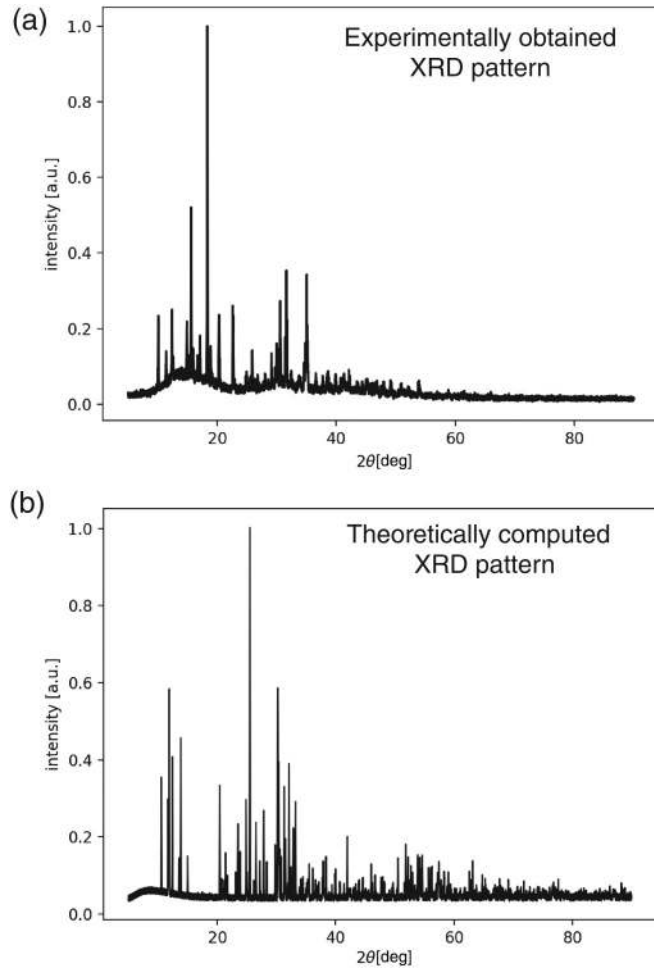


FIG. 1. Typical samples of powder XRD patterns. (a) XRD pattern for Borax  $\text{Na}_2[\text{B}_4\text{O}_5(\text{OH})_4] \cdot 8\text{H}_2\text{O}$  (obtained from the RRUFF database); (b) theoretically computed XRD pattern for  $\text{C}_2(\text{Si}_6\text{Cl}_{14})$  including noise and background. Both compounds belong to the space group 15.

Section IV shows how the theoretical training data is prepared, and Sec. V discusses all the results including the performance comparison between the convolutional and the dense network.

## II. X-RAY DIFFRACTION AND CRYSTAL SYMMETRIES

The goal of our machine learning task is to predict crystal symmetry (crystal system and space group) from powder XRD patterns. There are seven crystal systems and 230 space groups. In an XRD experiment, the photons scatter off the atoms in the crystal, and interfere and produce Bragg diffraction patterns which are collected by a detector. Since the sample is powdered, all crystal orientations are represented and hence the diffraction pattern shows rings centered about the beam axis, i.e., the intensity pattern  $S(2\theta)$  is a function of the scattering angle  $2\theta$  with respect to the beam axis. In Fig. 1, examples of such diffraction patterns are shown. Since the interference depends on the relative positions of the atoms within the crystal, information regarding the crystal symmetries is contained within the XRD pattern.

Conventional methods [34–39] for obtaining the space group from the XRD pattern generally involve performing some peak finding algorithm. Our objective is to train a neural network such that it learns the relevant features of the interference pattern in order to correctly predict the crystal symmetry without relying on preprocessing of the data to identify peaks. The XRD pattern, i.e., the Bragg peak positions in  $2\theta$ , depends on the wavelength of the photons. Generally, the patterns can be expressed in terms of scattering wave vectors  $q = 4\pi \sin(\theta)/\lambda$ . However, for direct comparison with Ref. [23], we fixed the wavelength at the copper  $K\alpha$  line ( $\lambda = 1.54 \text{ \AA}$ ) and hence we display our results as a function of  $2\theta$ . Since a neural network takes a finite number of inputs, we express the intensity function  $S(2\theta)$  as a vector. We set the range from  $5^\circ < 2\theta < 90^\circ$  with a spacing of  $0.01^\circ$ . In addition, we normalize the functions such that its largest value is one.

## III. NEURAL NETWORK AND TRAINING ALGORITHM

Artificial neural networks are variational approximations to arbitrary functions. Many variants and architectures exist. Here, we consider the so-called feed-forward neural networks (FFNN). They consist of a series of successively applied maps. Each map constitutes a “layer” of the network. By definition, the first layer is the input layer and the last layer is the output layer of the network.

Let  $\mathbf{v}_n$  be the output of layer  $n$  and define the input to the network to be  $\mathbf{v}_0$ . The output of layer  $n$  is then the input for layer  $n + 1$ . At each layer, we perform a transformation to go from one layer to the next. The most important layer consists of an affine map followed by a nonlinear function (or activation function)  $g_n$ , i.e.,

$$\mathbf{v}_n \rightarrow \mathbf{v}_{n+1} = g_n(\mathbf{W}_n \mathbf{v}_n + \mathbf{b}_n), \quad (1)$$

where  $\mathbf{W}_n$  is the so-called weight matrix and  $\mathbf{b}_n$  is the bias vector. When no further constraints are imposed on the weight matrix and bias vector, the layer is called a dense layer. In a so-called convolutional layer, by contrast, not all elements of the weight matrix can be freely chosen. There is a set of constraints which reduces the number of learnable parameters. This allows one to effectively use networks for much larger dimensions of the input vectors and to build both “deeper” and “wider” networks consisting of more layers. There are other types of layers with specific purpose, such as pooling layers and dropout layers [40].

An important freedom in the setup of the network is the choice of activation function  $g_n$ . For all layers apart from the final output layer, we use the rectified linear unit (ReLU) defined by

$$\text{ReLU}(z) = \max(0, z), \quad (2)$$

which is applied element wise to the vector  $\mathbf{v}_n$ . For the output layer, we use the SOFTMAX function given by

$$\text{SOFTMAX}(\mathbf{v}_n)_i = \frac{\exp(v_{n,i})}{\sum_j \exp(v_{n,j})}. \quad (3)$$

This SOFTMAX function is positive definite and normalized to 1 such that the output layer can be interpreted as a probability distribution.

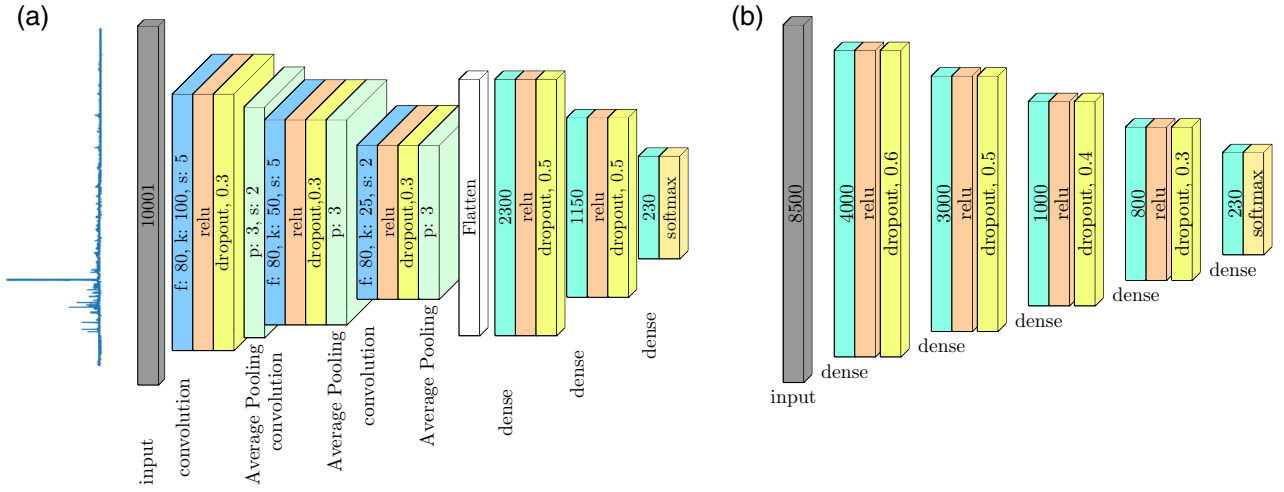


FIG. 2. Network architectures used for classifying space groups from powder XRD patterns. (a) Convolutional neural network as used in Ref. [23]. (b) Dense network. Blue layers represents convolutional layers with  $f$  filters of size  $k$  and stride  $s$ , green layers correspond to dense layers with the number of nodes given by the value within and yellow layers are dropout layers with the indicated dropout rate. The gray layers are the input layers with the number denoting the corresponding input size. The XRD patterns are zero padded to match the input size. Evaluated on the experimental samples from the RRUFF database, we achieved an accuracy of 42% for the convolutional network and an accuracy of 54% for the dense network.

In this paper, we consider two different network architectures, a convolutional network (this is the same architecture as in Ref. [23]) and a dense network. We use a one-hot encoding for representing the different possible classification categories. This means that the final layer of the network has as many nodes as there are categories for a certain classification of the input data, i.e., seven nodes if we are classifying crystal systems and 230 if we are classifying space groups. A fully confident classification into category  $i$  then corresponds to an output vector equal to the unit vector in  $i$  direction. More generally, the network's prediction is taken to be the category with the largest component of the output vector. The precise network architectures we used for space group classification are shown in Fig. 2. Notice that the final layer contains 230 nodes corresponding to different possible space groups. The networks used for classifying crystal systems are nearly identical apart from the final few layers since the output layer should have only seven nodes.

We now proceed to explain how the network is trained. For that, we use a labeled dataset  $\{(x, l_x)\}$  where  $x$  is the sample input (the XRD pattern) and  $l_x$  is the correct label (the space group or crystal system to which the pattern belongs). Let the network output distribution corresponding to the input  $x$  be given by  $y_x$ , and that corresponding to the correct label is defined via its components  $y_{x,i}^c = \delta_{i,l_x}$ . The network is trained by minimizing the following cost functions: the mean-squared-error or quadratic cost,

$$C_{\text{quad}} = \frac{1}{n} \sum_x |y_x - y_x^c|^2, \quad (4)$$

and the categorical cross entropy,

$$C_{\text{cross}} = -\frac{1}{n} \sum_x [y_x^c \cdot \ln y_x + (\mathbf{1} - y_x^c) \cdot \ln(\mathbf{1} - y_x)], \quad (5)$$

where the sum is over the training dataset. To minimize the cost function, we use a gradient decent type optimizer called adaptive moment estimation (Adam) [41] which proceeds as follows. We denote by  $\alpha_r$  the collection of all network parameters, i.e., all the entries of weight matrices and bias vectors, where  $r$  indexes all these quantities.

(1) At step  $t$ , compute the gradient of the cost function  $\mathcal{C}$  with respect to the network parameters,

$$g_{r;t} = \nabla_{\alpha_r} \mathcal{C}. \quad (6)$$

(2) Compute the decaying first and second moments of the past gradients.

$$\begin{aligned} m_{r;t} &= \beta_1 m_{r;t-1} + (1 - \beta_1) g_{r;t}, \\ v_{r;t} &= \beta_2 v_{r;t-1} + (1 - \beta_2) g_{r;t}^2, \end{aligned} \quad (7)$$

where  $\beta_1$  and  $\beta_2$  are the hyperparameters controlling the decay rate.

(3) Because these moments are initialized to zero ( $m_{r;0} = v_{r;0} = 0$ ), there is a bias towards zero. To counteract this, we define the bias-corrected moments,

$$\begin{aligned} \hat{m}_{r;t} &= \frac{m_{r;t}}{1 - \beta_1^t}, \\ \hat{v}_{r;t} &= \frac{v_{r;t}}{1 - \beta_2^t}. \end{aligned} \quad (8)$$

(4) Update the parameters of the network as

$$\alpha_r \rightarrow \alpha_r - \frac{\eta}{\sqrt{\hat{v}_{r;t}} + \epsilon} \hat{m}_{r;t}, \quad (9)$$

where  $\eta$  is the learning rate and  $\epsilon$  is a smoothing term to prevent division by zero.

We used the standard values for the hyperparameters of the Adam optimizer:  $\eta = 0.001$ ,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ , and  $\epsilon = 10^{-8}$ .

In order to speed up the training, instead of summing over the entire dataset in the cost functions  $\mathcal{C}$ , we can perform the sum over a randomly selected batch of samples. This introduces noise in the computation of the derivative. The smaller the batch size, the larger the noise. It is thus necessary to find a trade-off between speed and accuracy. However, it should be noted that a certain amount of noise can be useful to prevent the network from getting trapped in local minima.

In each iteration, a batch of samples from the training data is fed to the network and the parameters are updated according to the algorithm above. An epoch is one full training cycle of the machine learning algorithm such that that every sample in the training set is exposed. During the training, we alternate between the quadratic cost  $\mathcal{C}_{\text{quad}}$  on even epochs and the cross entropy  $\mathcal{C}_{\text{cross}}$  on odd epochs. The reason for this is that the categorical cross entropy helps to speed up convergence while the quadratic error is necessary for the network to produce a more meaningful output distribution. Without the quadratic error, the output of the dense network is always close to a one-hot encoded vector even for wrong predictions.

#### IV. DATA PREPARATION

The XRD patterns used for training the networks were computed theoretically from the crystal structure information obtained from the Inorganic Crystal Structure Database (ICSD) [26] which provides the crystallographic information file (CIF) of many inorganic substances. We removed duplicates (some crystal structures appeared more than once in the database) from the training data. With this restriction, a dataset of 1 28 404 samples was obtained.

The CIF contains the necessary information required to compute XRD patterns. Using the Python pymatgen library [42], we computed the theoretical XRD pattern in terms of peak heights and positions from the CIF. The peak heights and positions are then convolved with a Gaussian function of variable standard deviation  $\sigma$  to mimic the finite experimental resolution.

However, these theoretically computed patterns still lack counting statistics and background signals present in real experimental data. A network trained using only such data may interpret the experimental noise as Bragg peaks potentially causing a misclassification. To simulate statistical noise, we augment our XRD patterns with a random signal drawn from a uniform distribution. In addition, we add a background signal  $f(\theta)$  which is the sum of the following functions.

(1) Smooth step functions:

$$\begin{aligned} f_{\text{stepup}}(\theta) &= h \left( \frac{1}{1 + \exp[a(\theta - \theta_{\text{step}})]} \right), \\ f_{\text{stepdown}}(\theta) &= h \left( 1 - \frac{1}{1 + \exp[a(\theta - \theta_{\text{step}})]} \right), \end{aligned} \quad (10)$$

where  $h$ ,  $a$ , and  $\theta_s$  gives the height, steepness, and position of the step, respectively.

(2) Fourth-order polynomials,

$$f_p(\theta) = \left| \sum_{n=0}^4 a_n \theta^n \right|, \quad (11)$$

where the coefficient  $a_n$  are chosen randomly.

(3) To mimic enhanced background near the direct beam (small scattering angles  $2\theta$ ), but before the beam stop, we used

$$f_{\text{bump}}(\theta) = h(n\theta)^2 e^{-n\theta}, \quad (12)$$

where  $h$  is setting the magnitude and  $n$  is a measure of the steepness.

The exact details of how these background and noise parameters are chosen is given in the Appendix.

In Fig. 1, we compare the XRD pattern produced using the above procedure with a true experimental data from the RRUFF database. The noise and background signals in the two patterns are relatively similar suggesting that the method we used to create the data could help the network generalize better to experimental conditions. As a side note, it can be seen that although both crystals in Fig. 1 correspond to space group 15, their XRD patterns barely have any similarities that can be easily picked out by the naked eye. This illustrates the significant difficulty in the classification problem that the network faces.

The full dataset is then divided into three sets: test, validation, and training. The test and validation sets have a size of 7000 each, leaving us with 1 14 404 samples in the training set.

## V. RESULTS AND DISCUSSION

### A. Training

Using the dataset described in the previous section, we train the network architectures shown in Fig. 2. The evolution of the classification accuracies of the networks over both the test set and the RRUFF database (real experimental data) during the training are shown in Fig. 3. The accuracies were averaged over 10 trained networks initialized with a different random seed. For the final prediction, the output of the network was averaged over the ensemble of 10 networks. The final accuracies of the network ensembles are summarized in Table I. Figures 3(a) and 3(b) show the results for the crystal systems and space groups, respectively.

As expected, the networks are clearly more accurate in classifying the theoretical test data (which has the same noise and background structure as the training data) than the experimental data (which contains noise and background structures not present in the training data). This suggests that despite the addition of background and noise there are still important systematic differences between the theoretical and experimental data which may be one of the main obstacles for obtaining a higher classification accuracy on the experimental data. However, it is remarkable that this discrepancy is much larger in the convolutional network as compared to the dense network. In fact this discrepancy is so large that, even though the former had a higher accuracy over the test set, the RRUFF database is significantly better classified by the dense network. This means that the dense network generalizes better to imperfect data.

As an additional remark, it should be noted that the correct classification may not be possible, for instance, in the case of noncentrosymmetric crystals in which the experiment averages over twin domains. For completeness, we also restrict the classification to only centrosymmetric groups, but we do not see a significant improvement in the accuracies of the

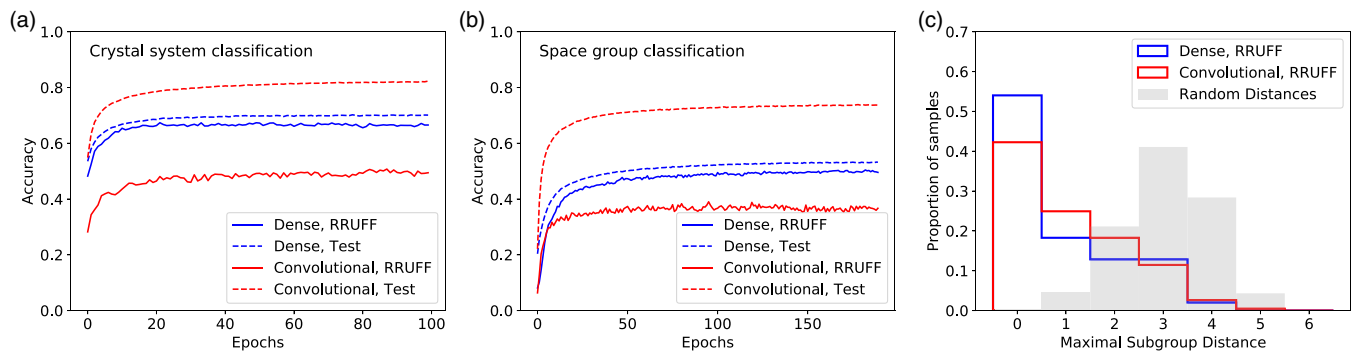


FIG. 3. Network classification accuracies. (a) and (b) The evolution of the network accuracies evaluated over the test set and the RRUFF database, which consists of experimentally obtained XRD patterns, for (a) the classification of crystal systems and (b) the classification of space groups. The accuracies are averaged over 10 networks initialized with a different random seeds. While the convolutional network performs better on the training data in both cases, the dense network generalizes better to the experimental RRUFF database. (c) Histogram of the distances between network’s prediction and correct space group classification. This distance is defined on the maximal subgroup graph.

networks. This is in part due to the fact that most of the samples both in the training and testing datasets are centrosymmetric since most crystals have this property. This is a reflection of the class imbalance problem. In order to better visualize this issue, a plot of the confusion matrix is included in the Appendix.

**B. Classification accuracy of individual space groups**

Next, we look at the classification accuracies of individual space groups. Since the distribution of space groups within the training set is highly nonuniform, we expect that the trained network would perform better at classifying the more common space groups. As can be seen in Fig. 4, this is indeed the case. Moreover, we also observed that the larger space groups (i.e., space groups with more symmetry elements) are more accurately classified. This is not surprising since with more symmetry elements there are more constraints on the structure factors giving rise to simpler XRD patterns with fewer Bragg peaks.

**C. Maximal subgroup distance**

To further elucidate the classification quality of the network, we consider the distance measure on the set of space groups to assess how far a wrong classification by the network is off the correct classification. This distance measure is defined through the concept of maximal subgroups. A maximal subgroup  $\mathcal{B}$  of  $\mathcal{A}$  is a proper subgroup such that no other proper subgroup  $\mathcal{C}$  strictly contains  $\mathcal{B}$ . This allows

us to construct a graph where the nodes represent the space group and two nodes are connected by an edge if one of the corresponding space groups is a maximal subgroup of the other. We can then define the maximal subgroup distance between two space groups to be the shortest path connecting their respective nodes. Intuitively, one may expect that  $\mathcal{A}$  and  $\mathcal{B}$  differ by a few symmetry elements such that their respective XRD patterns should be relatively similar and the network may confuse them more easily. Figure 3(b) shows the histogram of distances in the maximal subgroup graph for the network’s prediction over both the test set and the RRUFF database. It clearly illustrates that even when the network’s prediction is incorrect, it is far from random. It often lies

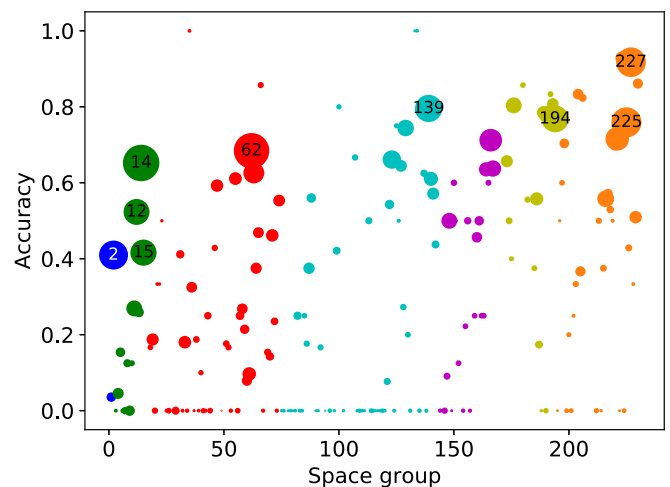


FIG. 4. Classification accuracy of individual space groups. The colors denote the corresponding crystal system and the area of the circle indicates its relative abundance in the training set. Space groups corresponding to larger indices tend to contain more symmetry elements. For instance, space group 1 contains no point group symmetries whereas space group 230 contains the full set of cubic symmetries. Depicted here are the accuracies for the theoretically obtained test set, because the RRUFF database is too small to provide meaningful statistics for individual space groups. (There are around 800 samples for 230 space groups in the RRUFF database.)

TABLE I. Obtained classification accuracies of the convolutional and dense networks for the theoretically computed test set data and for the experimental RRUFF data. In the last column, we restrict the classification to only centrosymmetric space groups.

	Crystal systems		Space groups		Centrosymmetric RRUFF
	Test set	RRUFF	Test set	RRUFF	
Convolutional	85%	56%	76%	42%	43%
Dense	73%	70%	57%	54%	59%

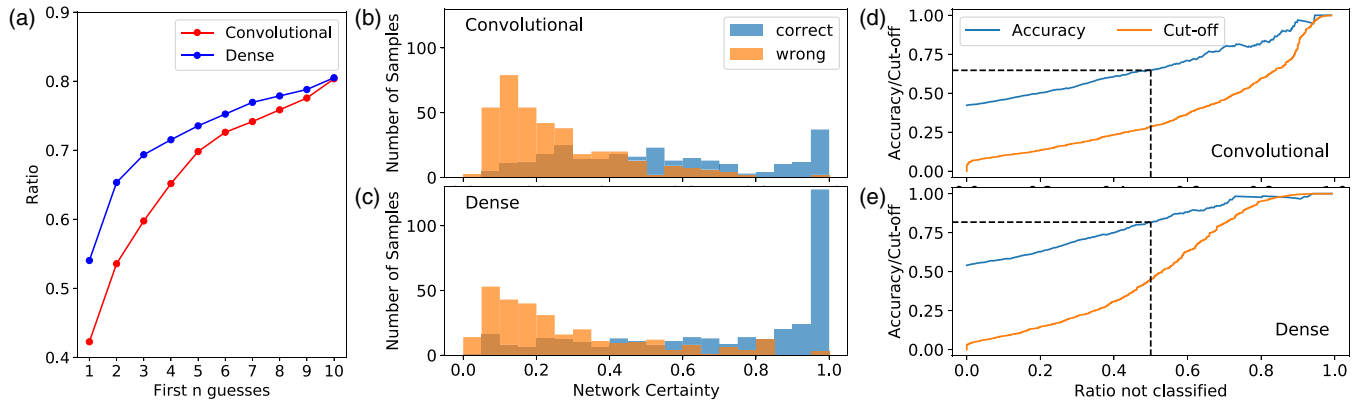


FIG. 5. (a) Ratio of samples whose corresponding space group is among the  $n$  highest outputs of the network. (b) and (c) Histograms showing the number of samples in the RRUFF database which are classified correctly or wrongly with a given certainty [defined in Eq. (13)]. One can see that when the network is more certain, it is also more often correct. (b) Convolutional network. (c) Dense network. (d) and (e) Depicts the accuracy (fraction of correctly classified samples on the RRUFF database) of the network if we restrict the network's certainty to be above a certain cutoff threshold. Given a XRD sample, if the network gives a certainty below the threshold, then the sample is ignored or unclassified. The orange line gives the ratio of unclassified samples as the cutoff is varied while the blue lines shows the accuracy of the network given a certain ratio of unclassified samples. The dotted lines indicate the accuracy when 50% of the samples are left unclassified. At this cutoff level, the accuracies are 65% and 82% for the (d) convolutional network and the (e) dense network, respectively.

closer to the correct prediction than one would expect from a random choice. This also suggests a possible alternative to interpret the classification output of the network: A space group prediction  $\mathcal{X}$  indicates with high probability that the corresponding crystal is in  $\mathcal{X}$  or in one of the neighboring space groups in the maximal subgroup graph.

#### D. Analysis of the full output distribution of the network

So far, we have only considered the space group/crystal system that corresponds to the maximum of the output vector  $\mathbf{y}$  as the classification result of the network. However, the network's output  $\mathbf{y}$  is in fact a probability distribution over the possible categories that contains much more information. We define the network's first prediction (this is the same as the network's prediction result) as the most probable category, the second prediction as the next most probable, and so on.

In Fig. 5(a), we show the first 10 predictions of neural networks evaluated on the test set and RRUFF database. As expected, we can see from the decreasing distance between consecutive data points that the first prediction is more often correct than the second prediction, the second prediction is more often correct than the third, and so on. This shows that there is additional useful information contained within the network's output on top of just its prediction which corresponds to its first prediction. Therefore, if we find that the network's prediction is wrong, the next most likely correct answer is indeed the second prediction. In other words, the probability distribution over the various categories as given by the network's output vector  $\mathbf{y}$  is a good prior.

#### E. Network certainty

Finally, we consider how to use network's certainty of its prediction to enhance the classification accuracies. Given

a network output  $\mathbf{y}$ , the certainty is then defined by the probability of the network's prediction, i.e.,

$$\text{certainty} = \max_i y_i, \quad (13)$$

where  $i$  runs over all space groups/crystal systems.

In Figs. 5(b) and 5(c), we show the classification certainties of both the convolutional and dense networks evaluated over the RRUFF dataset. The figures show that for the dense network, when the prediction is correct, the certainty is high. This result can be used to increase the accuracy of network classifications, if one allows the network to not classify inputs with low certainty, as follows: Let  $\mathbf{y}$  be the network's output for some input XRD pattern that is to be classified. If the network's certainty is above a certain threshold, we accept the classification. Otherwise we say that the network is uncertain and leave the input unclassified. Figures 5(d) and 5(e) show the result of such a scheme: Placing the certainty threshold at 0.45, we get a classification accuracy of around 82% for the dense network on the RRUFF database at the cost of having half of the data unclassified. The rationale behind such a tradeoff is that it is better to know that the network is uncertain about a (possibly low-quality) input rather than for the network to give a wrong classification.

It is imperative to use Fig. 5 for a comparison between the convolutional and the dense network. Whereas the dense network's prediction is often correct when it is certain, the correlation between certainty and correct classification is much weaker in the convolutional network. For instance, if we impose the same certainty threshold of 0.45 on the convolutional network, we would have around 69% unclassified data, and if we impose a threshold such that only half of the data is unclassified, the final classification accuracy is only 65%. This once again suggests that the convolutional network is not generalizing well to the experimental data.



## VI. CONCLUSION

We have trained neural networks for the purpose of classifying powder XRD patterns by the respective crystal system and space group of the crystal. The training was performed with theoretically computed training data based on real crystal structures. In order for the trained network to classify experimentally obtained data, we added noise and artificially constructed background signals to the training data.

In particular, we compared the performance of a convolutional network that was previously introduced in Ref. [23] for the same purpose and a deep dense neural network. We find that although the convolutional network classifies theoretical data much more accurately, it generalizes poorly to experimental data. This is consistent with the results in Ref. [23], where it was found that even though the network had above 80% accuracy on theoretical data, it failed on all of three experimental samples studied. We find that the deep dense network has a higher classification accuracy (54%) on experimental data than the convolutional network (42%).

Next, to better understand the quality of the network's prediction, we used the concept of maximal subgroups to provide a distance measure on the set of space groups. We found that even though the network might give a wrong prediction, this predicted space group is not random but instead lies close to the correct answer.

Finally, we analyzed the network's certainty to enhance the classification accuracy. By allowing the network to be undecided about a subset of the predictions, we were able to enhance the classification accuracy of the dense network (on experimental data) to 82% at the expense of leaving around half of the samples unclassified. This could be considered an improvement since it is helpful to know beforehand that a network's prediction is wrong. For the convolutional network, in contrast, we find that the certainty is not as good an indicator of correctness of the network's prediction.

Several routes for future improvements on our results present themselves and may lead to broad applications of machine learning techniques for crystallography. For one, the models we employ for artificial noise and background signals may be improved to increase the accuracy. Second, performant algorithms may be obtained by combining traditional and machine-learning methods, for instance, by using a neural network to identify the relevant peaks in the XRD pattern and then using standard crystallographic tools to classify the peaks. Finally, a growing experimental database of XRD samples may provide better, experimental training data.

## ACKNOWLEDGMENTS

We would like to thank Ruggero Frison, Antonio Cervellino, Denis Cheptikov, Ekaterina Pomjakushina, Pascal Puphal, and Ann-Christin Dippel for discussions as well as comments on the manuscript, and for sharing of their diffraction data. K.C. was supported by the European Unions Horizon 2020 research and innovation program (ERC-StG-Neupert-757867-PARATOP). J.C. acknowledges support from the Swiss National Science Foundation. The machine learning techniques were implemented using KERAS [43] with a TENSORFLOW backend [44].

## APPENDIX A: DATA PREPARATION

The XRD patterns used for training the networks were computed theoretically from the crystal structures information obtained from the Inorganic Crystal Structure Database [26]. We denote the theoretically computed XRD pattern by  $f_{\text{theory}}(\theta)$  which is defined on the domain  $\theta_{\min} < \theta < \theta_{\max}$  and takes values in the codomain  $0 \leq f_{\text{theory}}(\theta) \leq 1$ . However, these patterns still lack the noise and background signals present in real experimental data. A network trained using only such data may interpret the experimental noise as Bragg peaks potentially causing a misclassification. To introduce noise, we augment our XRD patterns with a random signal drawn from a uniform distribution  $\mathcal{U}(0.002, 0.02)$  between 0.002 and 0.02.

Next, to simulate a background we add a signal which is composed of four different functions. These four functions are selected with a 50% probability each, i.e., not all four functions have to appear in our background signal. We also define a threshold  $T$  given by

$$T = \frac{0.1}{\text{number of different background functions chosen}},$$

which controls the amplitude of each background function. For convenience, we also define the relative angle  $\theta_{\text{rel}} = \frac{\theta - \theta_{\min}}{\theta_{\max} - \theta_{\min}}$ . The four background functions are given by

(1) Smooth step functions,

$$\begin{aligned} f_{\text{stepup}}(\theta) &= h \left( \frac{1}{1 + \exp[a(\theta_{\text{rel}} - \theta_{\text{rel,step}})]} \right), \\ f_{\text{stepdown}}(\theta) &= h \left( 1 - \frac{1}{1 + \exp[a(\theta_{\text{rel}} - \theta_{\text{rel,step}})]} \right), \end{aligned} \quad (\text{A1})$$

where  $h$ ,  $a$ , and  $\theta_{\text{rel,step}}$  gives the height, steepness, and position of the step, respectively. These parameters are chosen as follows:  $h$  is drawn a truncated normal distribution  $h \in [0, T]$  with mean  $\mu = T/3$  and standard deviation  $\sigma = T/7$ ,  $a$  is sampled from a uniform distribution  $\mathcal{U}(10, 60)$ , and  $\theta_{\text{rel,step}}$  for the step-up (step-down) function is taken from a uniform distribution, with a width of  $W = 1/7$ , at the left edge (right edge) of the domain, i.e.,  $\mathcal{U}(0, W)$  for the step-up function and  $\mathcal{U}(1 - W, 1)$  for the step-down function.

(2) Polynomials (up to fourth order),

$$f_p(\theta) = \left| \sum_{n=0}^{n_{\max}} a_n \theta_{\text{rel}}^n \right|, \quad (\text{A2})$$

where  $n_{\max}$  is a random integer between 0 and 4. The coefficient  $a_n$  are chosen according to

$$a_n = \begin{cases} 0 & \text{with probability } 0.5 \\ \frac{3T}{2(n_{\max}+1)} \mathcal{U}(-1, 1) & \text{with probability } 0.5 \end{cases} \quad (\text{A3})$$

(3) Bumps near the left edge of the XRD pattern (near  $\theta_{\min}$ ), to mimic the enhanced background near the direct beam,

$$f_{\text{bump}}(\theta) = h(n\theta_{\text{rel}})^2 e^{-n\theta_{\text{rel}}}, \quad (\text{A4})$$

where  $h$  is the height of the bump and  $n$  is a measure of the steepness.  $h$  is drawn from a truncated normal distribution  $h \in [0, 3T/5]$  with mean  $\mu = 2T/5$  and standard deviation  $\sigma = 3T/35$ , and  $n$  is taken from a uniform distribution  $\mathcal{U}(40, 70)$ .

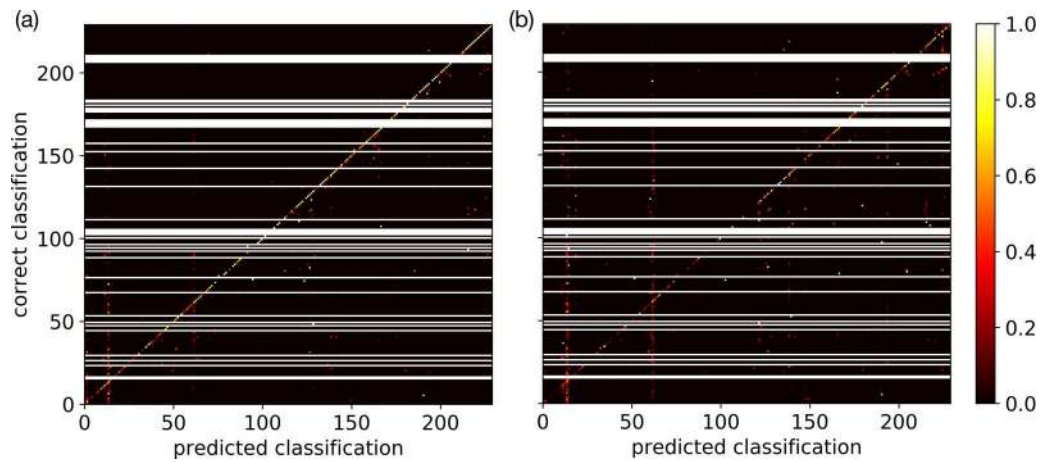


FIG. 6. Confusion matrices are evaluated only on the theoretical test data (7000 samples), because the experimental RRUFF database is too small to provide meaningful statistics for individual space groups. (There are around 800 samples for 230 space groups in the RRUFF database.) The white strips in the figure corresponds to space groups which are not present among the 7000 test samples. (a) Convolutional network. (b) Dense network.

The full augmented XRD pattern is then given by

$$f(\theta) = f_{\text{theory}}(\theta) + f_{\text{stepup}}(\theta) + f_{\text{stepdown}}(\theta) + f_p(\theta) + f_{\text{bump}}(\theta) + \text{noise}, \quad (\text{A5})$$

where the parameters of the individual functions are chosen independently according to the procedure described above.

## APPENDIX B: CONFUSION MATRIX

The confusion matrix is a common method to visualize the performance of a classifier. For a classification problem with

$N$  possible classes, the confusion matrix  $\mathcal{C}$  is a  $N$  by  $N$  matrix with coefficients defined by

$$C_{ij} = \frac{\text{Number of class } i \text{ samples classified as class } j}{\text{Number of class } i \text{ samples}}. \quad (\text{B1})$$

Defined as such, a perfect classifier would have a confusion matrix corresponding to the identity. In Fig. 6, we show a color plot of the confusion matrices corresponding to convolutional and dense networks, evaluated on a testing dataset of 7000 samples.

- 
- [1] H. M. Rietveld, *J. Appl. Crystallogr.* **2**, 65 (1969).  
 [2] J. Rodríguez-Carvajal, *Phys. B: Condens. Matter* **192**, 55 (1993).  
 [3] M. R. Eskildsen, E. M. Forgan, and H. Kawano-Furukawa, *Rep. Prog. Phys.* **74**, 124504 (2011).  
 [4] J. C. Kendrew, G. Bodo, H. M. Dintzis, R. G. Parrish, H. Wyckoff, and D. C. Phillips, *Nature (London)* **181**, 662 (1958).  
 [5] R. E. Franklin and R. G. Gosling, *Nature (London)* **171**, 740 (1953).  
 [6] J. D. Watson and F. H. C. Crick, *Nature (London)* **171**, 737 (1953).  
 [7] J. Chipman, *Metall. Mater. Trans. B* **3**, 55 (1972).  
 [8] C. M. Song, S. J. Lim, and J. C. Tong, *Briefings Bioinf.* **10**, 579 (2009).  
 [9] V. Stanev, C. Oses, A. G. Kusne, E. Rodriguez, J. Paglione, S. Curtarolo, and I. Takeuchi, *npj Comput. Mater.* **4**, 29 (2018).  
 [10] Y. Zhang, A. Mesaros, K. Fujita, S. Edkins, M. Hamidian, K. Ch'ng, H. Eisaki, S. Uchida, J. Davis, E. Khatami, and E. Kim, [arXiv:1808.00479](https://arxiv.org/abs/1808.00479).  
 [11] G. Pilia, A. Mannodi-Kanakkithodi, B. Uberuaga, R. Ramprasad, J. Gubernatis, and T. Lookman, *Sci. Rep.* **6**, 19375 (2016).  
 [12] O. Isayev, C. Oses, C. Toher, E. Gossett, S. Curtarolo, and A. Tropsha, *Nat. Commun.* **8**, 15679 (2017).  
 [13] A. Seko, T. Maekawa, K. Tsuda, and I. Tanaka, *Phys. Rev. B* **89**, 054303 (2014).  
 [14] P. V. Balachandran, J. Theiler, J. M. Rondinelli, and T. Lookman, *Sci. Rep.* **5**, 13285 (2015).  
 [15] G. Carleo and M. Troyer, *Science* **355**, 602 (2017).  
 [16] K. Ch'ng, J. Carrasquilla, R. G. Melko, and E. Khatami, *Phys. Rev. X* **7**, 031038 (2017).  
 [17] E. P. Van Nieuwenburg, Y.-H. Liu, and S. D. Huber, *Nat. Phys.* **13**, 435 (2017).  
 [18] J. Carrasquilla and R. G. Melko, *Nat. Phys.* **13**, 431 (2017).  
 [19] G. Torlai and R. G. Melko, *Phys. Rev. Lett.* **119**, 030501 (2017).  
 [20] P. B. Wigley, P. J. Everitt, A. van den Hengel, J. Bastian, M. A. Sooriyabandara, G. D. McDonald, K. S. Hardman, C. Quinlivan, P. Manju, C. C. Kuhn *et al.*, *Sci. Rep.* **6**, 25890 (2016).  
 [21] P. Baldi, P. Sadowski, and D. Whiteson, *Nat. Commun.* **5**, 4308 (2014).  
 [22] F. Schindler, N. Regnault, and T. Neupert, *Phys. Rev. B* **95**, 245134 (2017).  
 [23] W. B. Park, J. Chung, J. Jung, K. Sohn, S. P. Singh, M. Pyo, N. Shin, and K.-S. Sohn, *IUCrJ* **4**, 486 (2017).

- [24] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis, *Science* **362**, 1140 (2018).
- [25] A. Ziletti, D. Kumar, M. Scheffler, and L. M. Ghiringhelli, *Nat. Commun.* **9**, 2775 (2018).
- [26] A. Belsky, M. Hellenbrandt, V. L. Karen, and P. Luksch, *Acta Crystallogr. Sec. B* **58**, 364 (2002).
- [27] R. T. Downs and M. Hall-Wallace, *Am. Mineral.* **88**, 247 (2003).
- [28] S. Gražulis, D. Chateigner, R. T. Downs, A. F. T. Yokochi, M. Quirós, L. Lutterotti, E. Manakova, J. Butkus, P. Moeck, and A. Le Bail, *J. Appl. Crystallogr.* **42**, 726 (2009).
- [29] S. Gražulis, A. Daškevič, A. Merkys, D. Chateigner, L. Lutterotti, M. Quirós, N. R. Serebryanaya, P. Moeck, R. T. Downs, and A. Le Bail, *Nucleic Acids Res.* **40**, D420 (2012).
- [30] A. Merkys, A. Vaitkus, J. Butkus, M. Okulič-Kazarinas, V. Kairys, and S. Gražulis, *J. Appl. Crystallogr.* **49** (2016).
- [31] S. Gražulis, A. Merkys, A. Vaitkus, and M. Okulič-Kazarinas, *J. Appl. Crystallogr.* **48**, 85 (2015).
- [32] A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, and K. A. Persson, *APL Mater.* **1**, 011002 (2013).
- [33] B. Lafuente, R. Downs, H. Yang, and N. Stone, in *Highlights in Mineralogical Crystallography*, edited by T. Armbruster and R. Danisi (Walter de Gruyter GmbH, Berlin, 2016), pp. 1–29.
- [34] P.-E. Werner, L. Eriksson, and M. Westdahl, *J. Appl. Crystallogr.* **18**, 367 (1985).
- [35] J. Visser, *J. Appl. Crystallogr.* **2**, 89 (1969).
- [36] A. Boulfif and D. Louër, *J. Appl. Crystallogr.* **24**, 987 (1991).
- [37] A. Le Bail, *Powder Diffr.* **19**, 249 (2004).
- [38] A. Altomare, M. Camalli, C. Cuocci, C. Giacovazzo, A. Moliterni, and R. Rizzi, *J. Appl. Crystallogr.* **42**, 1197 (2009).
- [39] M. A. Neumann, *J. Appl. Crystallogr.* **36**, 356 (2003).
- [40] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, Cambridge, 2016).
- [41] D. Kingma and J. Ba, [arXiv:1412.6980](https://arxiv.org/abs/1412.6980).
- [42] S. P. Ong, W. D. Richards, A. Jain, G. Hautier, M. Kocher, S. Cholia, D. Gunter, V. L. Chevrier, K. A. Persson, and G. Ceder, *Comput. Mater. Sci.* **68**, 314 (2013).
- [43] F. Chollet *et al.*, KERAS, <https://keras.io>.
- [44] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, Computer code TENSORFLOW: Large-scale machine learning on heterogeneous systems (2015); software available from, <http://www.tensorflow.org>.