Acta Cryst A FOUNDATIONS ADVANCES

ISSN 2053-2733

Using a machine learning approach to determine the space group of a structure from the atomic pair distribution function

Chia-Hao Liu,[a] Yunzhe Tao,[a] Daniel Hsu,[b] Qiang Du[a] and Simon J. L. Billinge[a,c]*

[a]Department of Applied Physics and Applied Mathematics, Columbia University, New York, New York, 10027, USA, [b]Department of Computer Science, Columbia University, New York, New York, 10027, USA, and [c]Condensed Matter Physics and Materials Science Department, Brookhaven National Laboratory, Upton, New York, 11973, USA. *Correspondence e-mail: sb2896@columbia.edu

Received 26 February 2019
Accepted 24 April 2019

Edited by A. Altomare, Institute of Crystallography - CNR, Bari, Italy

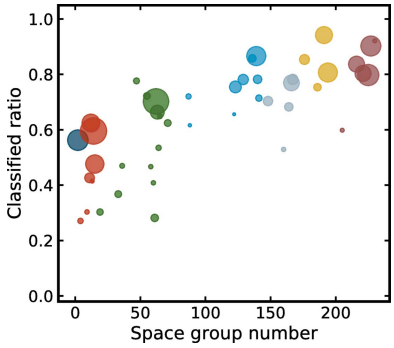
Keywords: pair distribution function; space groups; convolutional neural network; machine learning.

A method is presented for predicting the space group of a structure given a calculated or measured atomic pair distribution function (PDF) from that structure. The method utilizes machine learning models trained on more than 100 000 PDFs calculated from structures in the 45 most heavily represented space groups. In particular, a convolutional neural network (CNN) model is presented which yields a promising result in that it correctly identifies the space group among the top-6 estimates 91.9% of the time. The CNN model also successfully identifies space groups for 12 out of 15 experimental PDFs. Interesting aspects of the failed estimates are discussed, which indicate that the CNN is failing in similar ways as conventional indexing algorithms applied to conventional powder diffraction data. This preliminary success of the CNN model shows the possibility of model-independent assessment of PDF data on a wide class of materials.

1. Introduction

Crystallography is used to determine crystal structures from diffraction patterns (Giacovazzo, 1999), including patterns from powdered samples (Pecharsky & Zavalij, 2005). The analysis of single-crystal diffraction is the most direct approach for solving crystal structures. However, powder diffraction becomes the best option when single crystals with desirable size and quality are not available.

A crystallographic structure solution makes heavy use of symmetry information to succeed. The first step is to determine the unit cell and space group of the underlying structure. Information about this is contained in the positions (and characteristic absences) of Bragg peaks in the diffraction pattern. This process of determining the unit cell and space group of the structure is known as 'indexing' the pattern (Giacovazzo, 1999). Indexing is inherently challenging for powder diffraction due to the loss of explicit directional information in the pattern, which is the result of projecting the data from three dimensions into a one-dimensional pattern (de Wolff, 1957; Mighell & Santoro, 1975). However, there are a number of different algorithms available that work well in different situations (Visser, 1969; Coelho, 2003; Boultif & Louër, 2004; Altomare, Campi *et al.*, 2009). Once the unit-cell information is determined, an investigation on systematic absences of diffraction peaks is carried out to identify the space group. Various methods for determining space-group information, based on either statistical or brute-force searches, have been used (Neumann, 2003; Markvardsen *et al.*, 2008; Altomare, Camalli *et al.*, 2009; Coelho, 2017).



The problem is even more difficult when the structural correlations only extend on nanometre length scales as crystallography breaks down (Billinge & Levin, 2007). In this case progress can be made using atomic pair distribution function (PDF) methods for structure refinements (Proffen *et al.*, 2005; Egami & Billinge, 2012; Choi *et al.*, 2014; Zobel *et al.*, 2015; Keen & Goodwin, 2015). PDFs may also be used for studying structures of bulk materials.

There has been some success in using the PDF for structure solution (Juhás *et al.*, 2006, 2010; Billinge *et al.*, 2018; Cliffe *et al.*, 2010). However, a major challenge for PDF structure solution is that, unlike the powder diffraction case, a peak in the PDF simply indicates a characteristic distance existing in the structure but gives no overall information about the underlying unit cell (Egami & Billinge, 2012). Therefore, the symmetry information cannot be inferred by the traditional indexing protocols that are predicated on the crystallography. Being able to determine the symmetry information based on the PDF will lead to more possibilities of solving structures from a wider class of materials.

Recently, machine learning (ML) has emerged as a powerful tool in different fields, such as in image classification (Krizhevsky *et al.*, 2012) and speech recognition (Hinton *et al.*, 2012). Moreover, ML models even outperform a human in cases such as image classifications (He *et al.*, 2015) and the game of Go (Silver *et al.*, 2017). ML provides a platform for exploring the predictive relationship between the input and output of a problem, given a considerable amount of data is supplied for an ML model to 'learn'. We know that the symmetry information is present in the powder diffraction pattern, and that the PDF is simply a Fourier transform of that pattern. We therefore reason that the symmetry information survives in the PDF though we do not know explicitly how it is encoded. We can qualitatively deduce that a higher-symmetry structure, such as cubic, will produce a lower density of PDF peaks than a lower-symmetry structure such as tetragonal. However, to date, there has not been a theory for identifying the space group directly, given the PDF. Here we attempt to see whether an ML algorithm can be trained to recognize the space group of the underlying structure, given a PDF as input. We note a recent paper that describes an attempt to determine the space group from a powder diffraction pattern (Park *et al.*, 2017). In this case a promising accuracy of 81% was obtained in determining the space group from simulated data, but the convolutional neural network model they used was not able to determine the space group from experimental data selected in their work.

To prepare data for training an ML model, we compute PDFs from 45 space groups, totaling 101 802 structures, deposited in the Inorganic Crystal Structure Database (ICSD) (Belsky *et al.*, 2002). The space groups chosen were the most heavily represented, accounting for more than 80% of known inorganic compounds (Urusov & Nadezhina, 2009).

The first ML model we tried was logistic regression (LR), which is a rather simple ML model. Although quite successful, we explored a more sophisticated ML model, a convolutional neural network (CNN). The CNN model outperforms the LR model by 15%, reaching an accuracy of 91.9% for obtaining the correct space group in the top-6 predicted results on the testing set. In particular, the CNN showed a significant improvement over LR in classifying challenging cases such as structures with lower symmetry.

The CNN model is also tested on experimental PDFs where the underlying structures are known but the data are subject to experimental noise and collected under various instrumental conditions. High accuracy in determining space groups from experimental PDFs was also demonstrated.

2. The PDF method

The experimental PDF, denoted $G(r)$, is the truncated Fourier transform of the total scattering structure function, $F(Q) = Q[S(Q) - 1]$ (Farrow & Billinge, 2009),

$$G(r) = \frac{2}{\pi}\int_{Q_{\min}}^{Q_{\max}} F(Q)\sin(Qr)\,\mathrm{d}Q, \tag{1}$$

where Q is the magnitude of the scattering momentum. The structure function, $S(Q)$, is extracted from the Bragg and diffuse components of the powder diffraction intensity. For elastic scattering, $Q = 4\pi\sin(\theta)/\lambda$, where λ is the scattering wavelength and 2θ is the scattering angle. In practice, values of $Q_{\min}$ and $Q_{\max}$ are determined by the experimental setup and $Q_{\max}$ is often reduced below the experimental maximum to eliminate noisy data from the PDF since the signal-to-noise ratio becomes unfavorable in the high-Q region. The value of $Q_{\max}$ is also known to be a dominant factor for the termination ripples introduced in the truncated Fourier transform (Peterson *et al.*, 2003).

The PDF gives the scaled probability of finding two atoms in a material at distance r apart and is related to the density of atom pairs in the material (Egami & Billinge, 2012). For a macroscopic scatterer, $G(r)$ can be calculated from a known structure model according to

$$G(r) = 4\pi r[\rho(r) - \rho_0], \tag{2}$$

$$\rho(r) = \frac{1}{4\pi r^2 N}\sum_i\sum_{j\neq i}\frac{b_i b_j}{\langle b\rangle^2}\delta(r - r_{ij}). \tag{3}$$

Here, ρ_0 is the atomic number density of the material and $\rho(r)$ is the atomic pair density, which is the mean weighted density of neighbor atoms at distance r from an atom at the origin. The sums in $\rho(r)$ run over all atoms in the sample, b_i is the scattering factor of atom i, $\langle b\rangle$ is the average scattering factor, and r_{ij} is the distance between atoms i and j.

3. Machine learning experiments

ML is centered around the idea of exploring the predictive but oftentimes implicit relationship between inputs and outputs of a problem. By feeding a considerable amount of input and output pairs (training set) to a learning algorithm, we hope to arrive at a prediction model which is a good approximation to

Table 1
Space group and corresponding number of entries considered in this study.

Space group (No.)	No. of entries
$P\bar{1}$ (2)	4615
$P2_1$ (4)	581
Cc (9)	489
$P2_1/m$ (11)	1247
$C2/m$ (12)	3529
$P2/c$ (13)	442
$P2_1/c$ (14)	7392
$C2/c$ (15)	3704
$P2_12_12_1$ (19)	701
$Pna2_1$ (33)	743
$Cmc2_1$ (36)	525
$Pmmm$ (47)	646
$Pbam$ (55)	745
$Pnnm$ (58)	477
$Pbcn$ (60)	478
$Pbca$ (61)	853
$Pnma$ (62)	6930
$Cmcm$ (63)	2249
$Cmca$ (64)	575
$Cmmm$ (65)	513
$Immm$ (71)	754
$I4/m$ (87)	569
$I4_1/a$ (88)	397
$I\bar{4}2d$ (122)	373
$P4/mmm$ (123)	1729
$P4/nmm$ (129)	1376
$P4_2/mnm$ (136)	870
$I4/mmm$ (139)	4028
$I4/mcm$ (140)	1026
$I4_1/amd$ (141)	700
$R\bar{3}$ (148)	1186
$R3m$ (160)	482
$P\bar{3}m1$ (164)	1005
$R\bar{3}m$ (166)	2810
$R\bar{3}c$ (167)	1390
$P6_3/m$ (176)	1289
$P6_3mc$ (186)	849
$P6/mmm$ (191)	3232
$P6_3/mmc$ (194)	3971
$Pa\bar{3}$ (205)	447
$F\bar{4}3m$ (216)	2893
$Pm\bar{3}m$ (221)	2933
$Fm\bar{3}m$ (225)	4860
$Fd\bar{3}m$ (227)	4382
$Ia\bar{3}d$ (230)	455
Total	101 802

the underlying relationship between the inputs and outputs. If the exact form of the output is available, either discrete or continuous, before the training step, the problem is categorized as 'supervised learning' in the context of ML. The space-group determination problem discussed in this paper also falls into the supervised learning category. In the language of ML, the inputs are often denoted as 'features' of the data and the outputs are usually called the 'labels'. Both inputs and outputs could be a scalar or a vector. After learning, the prediction model is then tested against a set of input and output pairs which have not been seen by the training algorithm (the so-called testing set) in order to independently validate the performance of the prediction model.

In the context of the space-group determination problem, the input that we want to interrogate is PDF data. We can select any feature or features from the data, for example the feature we choose could be the PDF itself. The label is the space group of the structure that gave rise to the PDF. The database we will use to train our model is a pool of known structures. In particular, we choose all the known structures from the 45 most heavily represented space groups in the ICSD, which accounts for 80% of known inorganic compounds (Urusov & Nadezhina, 2009). These were further pruned to remove duplicate entries (same composition *and* same structure). The space groups considered and the number of unique structures in each space group are reproduced in Table 1.

Table 2
Parameters used to calculate PDFs from atomic structures.

ADP stands for isotropic atomic displacement parameter. All parameters follow the same definitions as in Farrow *et al.* (2007).

Parameter	Value
$r_{\rm min}$ (Å)	1.5
$r_{\rm max}$ (Å)	30.0
$Q_{\rm min}$ (Å^{-1})	0.5
$Q_{\rm max}$ (Å^{-1})	23.0
$r_{\rm grid}$ (Å)	$\pi/Q_{\rm max}$
ADP (Å^2)	0.008
$Q_{\rm damp}$ (Å^{-1})	0.04
$Q_{\rm broad}$ (Å^{-1})	0.01

We then computed the PDF from each of 101 802 structures. The parameters capturing finite Q range and instrumental conditions are reproduced in Table 2. Those parameters are chosen such that they are close to the values that are practically attainable at most synchrotron facilities. With the $r_{\rm grid}$ and r range reported in Table 2, each computed PDF is a 209×1 vector. Depending on the atom types in the compounds, the amplitude of the PDF may vary drastically, which is inherently problematic for most ML algorithms (James *et al.*, 2013). To avoid this problem, we determine a normalized PDF, $\mathbf{X}$, defined according to

$$\mathbf{X} = \frac{G(r) - \min(G)}{\max(G) - \min(G)}, \tag{4}$$

where $\min(G)$ and $\max(G)$ mean taking the minimum and maximum value of the target PDF function, $G(r)$, respectively. Since $\min(G)$ is always a negative number for the reduced PDF, $G(r)$, that we compute from the structure models, this definition results in the value of $\mathbf{X}$ always ranging between 0 and 1. An example of $\mathbf{X}$ from $Li_{18}Ta_6O_{24}$ (space group $P2/c$) is shown in Fig. 1(*a*).

For our learning experiments, we randomly select 80% of the data entries from each space group as the training set and reserve the remaining 20% of data entries as the testing set.

All learning experiments were carried out on one or multiple computation nodes of the Habanero shared high-performance cluster (HPC) at Columbia University. Each computation node consists of 24 cores of CPUs (Intel Xeon Processor E5-2650 v4), 128 GB memory and two GPUs (Nvidia K80 GPUs).

3.1. Space-group determination based on logistic regression (LR) model

We start our learning experiment with a rather simple model, LR. In the setup of the LR model the probability of a given feature being classified as a particular space group is parametrized by a 'logistic function' (Hastie *et al.*, 2009). Forty-five space groups are considered in our study; therefore there are the same number of logistic functions, each with a set of parameters left to be determined. Since the space-group label is known for each data entry in the training set, the learning algorithm is then used to find an optimized set of parameters for each of the 45 logistic functions such that the overall probability of determining the correct space group on all training data is maximized. As a common practice, we also include 'regularization' (Hastie *et al.*, 2009) to reduce overfitting in the trained model. The regularization scheme chosen in our implementation is 'elastic net' which is known for encouraging sparse selections on strongly correlated variables (Zou & Hastie, 2005). Two hyperparameters α and Λ are introduced under the context of our regularization scheme. The explicit definition of these two parameters is presented in Appendix *A*. Our LR model is implemented through *scikit-learn* (Pedregosa *et al.*, 2011). The optimum α, Λ for our LR model is determined by cross-validation (Hastie *et al.*, 2009) in the training stage.

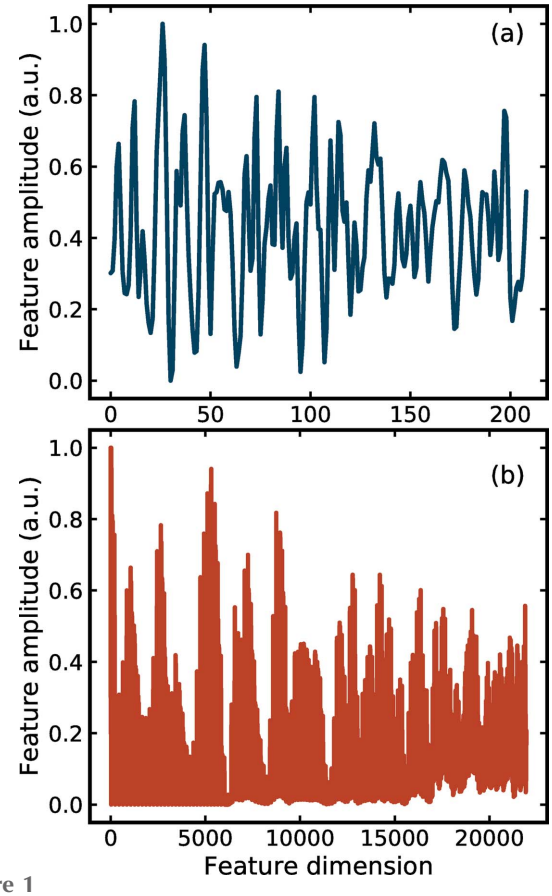


Figure 1
Example of (*a*) normalized PDF $\mathbf{X}$ and (*b*) its quadratic form $\mathbf{X}^2$ of compound $Li_{18}Ta_6O_{24}$ (space group $P2/c$).

The best LR model with $\mathbf{X}$ as the input yields an accuracy of 20% at $(\alpha, \Lambda) = (10^{-5}, 0.75)$. This result is better than a random guess from 45 space groups (2%) but is still far from satisfactory. We reason that the symmetry information depends not on the absolute value of the PDF peak positions, which depend on specifics of the chemistry, but on their relative positions. This information may be more apparent in an autocorrelation of the PDF with itself, which is a quadratic feature in ML language. Our quadratic feature, $\mathbf{X}^2$, is defined as

$$\mathbf{X}^2 = \{X_i X_j \,|i, j = 1, 2, \ldots d, j > i\} \tag{5}$$

where d is the dimension of $\mathbf{X}$ and $\mathbf{X}^2$ is a vector of dimension $\{[d(d-1)]/2\} \times 1$. An example of the quadratic feature from $Li_{18}Ta_6O_{24}$ (space group $P2/c$) is shown in Fig. 1(*b*).

The best LR model with $\mathbf{X}^2$ as the input yields an accuracy of 44.5% at $(\alpha, \Lambda) = (10^{-5}, 1.0)$. This is much better than for the linear feature, but still quite low. However, the goal of the space-group determination problem is to find the right space group, not necessarily to have it returned in the top position in a rank-ordered list of suggestions. We therefore define alternative accuracy (A_6) that allows the correct space group to appear at any position in the top-6 space groups returned by the model. The values of A_i $(i = 1, 2, \ldots 6)$ and their first discrete differences $\Delta A_i = A_i - A_{i-1}$ $(i = 2, 3, \ldots, 6)$ of our best LR model are shown in Fig. 2. We observed a more than 10% improvement in the alternative accuracy after considering top-2 predictions from the LR model (ΔA_2) and the improvement (ΔA_i) diminishes monotonically when more predictions are considered, as expected. A top-6 estimate yields a good accuracy (77%) and this is still a small enough number of space groups that could be tested manually in any structure determination.

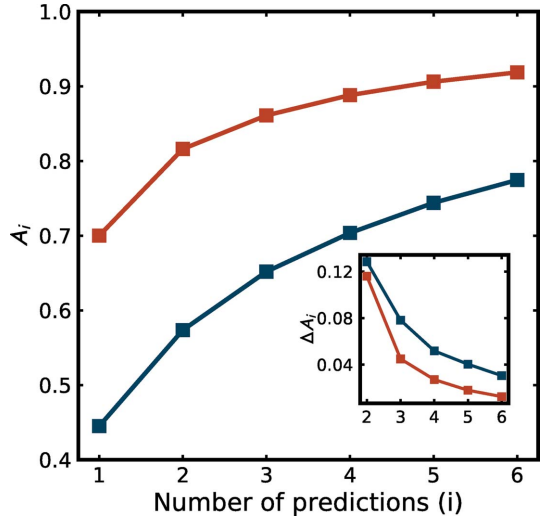


Figure 2
Accuracy in determining space group when top-i predictions are considered (A_i). The inset shows the first discrete differences ($\Delta A_i = A_i - A_{i-1}$) when i predictions are considered. Blue represents the result of the logistic regression model with $\mathbf{X}^2$ and red is the result from the convolutional neural network model.

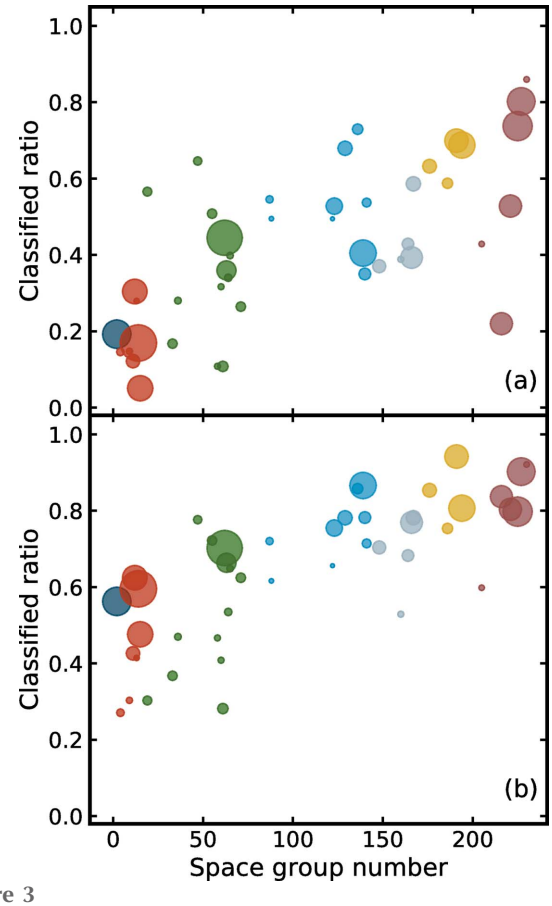


Figure 3
The ratio of correctly classified structures versus space-group number from (*a*) the logistic regression model (LR) with quadratic feature $\mathbf{X}^2$ and (*b*) the convolutional neural network (CNN) model. Marker size reflects the relative frequency of the space group in the training set. Markers are color coded with corresponding crystal systems [triclinic (dark blue), monoclinic (orange), orthorhombic (green), tetragonal (blue), trigonal (gray), hexagonal (yellow) and cubic (dark red)].

The ratio of correctly classified structures versus space-group number is shown Fig. 3(*a*).

The space-group numbering follows standard convention (Hahn, 2002). Higher space-group number means a more symmetric structure and we find, in general, the LR model yields a decent performance in predicting space groups from structures with high symmetry but it performs poorly on classifying low-symmetry structures.

3.2. Space-group determination based on the convolutional neural network (CNN)

The result from the linear ML model (LR) is promising, prompting us to move to a more sophisticated deep learning model. Deep learning models (LeCun *et al.*, 2015; Goodfellow *et al.*, 2016) have been successfully applied to various fields, ranging from computer vision (He *et al.*, 2016; Krizhevsky *et al.*, 2012; Radford *et al.*, 2015), natural language processing (Bahdanau *et al.*, 2014; Sutskever *et al.*, 2014; Kim, 2014) to material science (Ramprasad *et al.*, 2017; Ziletti *et al.*, 2018). In particular, we sought to use a CNN (Lecun *et al.*, 1998).

The performance of a CNN depends on the overall architecture as well as the choice of hyperparameters such as the size of kernels, the number of channels at each convolutional layer, the pooling size and the dimension of the fully connected (FC) layer (Goodfellow *et al.*, 2016). However there is no well-established protocol for selecting these parameters, which is a largely trial-and-error effort for any given problem. We build our CNN by tuning hyperparameters and validating the performance on the testing data, which is just 20% of the total data.

The resulting CNN built for the space-group determination problem is illustrated in Fig. 4.

The input PDF is a one-dimensional signal sequence of dimension $209 \times 1 \times 1$. We first apply a convolution layer of 256 channels with kernel size 32×1 to extract the first set of feature maps (Lecun *et al.*, 1998) of dimension $209 \times 1 \times 256$. It has been shown that applying a nonlinear activation function to each output improves not only the ability of a model to learn complex decision rules but also the numerical stability during the optimization step (LeCun *et al.*, 2015). We chose rectified linear unit (ReLU) (Dahl *et al.*, 2013) as our activation function for the network. After the first convolution layer, we apply a 64-channel kernel of size 32×1 to the first feature map and generate the second set of feature maps of dimension $209 \times 1 \times 64$. Similar to the first convolution layer, the second feature map is also activated by ReLU. This is followed by a max-pooling layer (Jarrett *et al.*, 2009) of size 2, which is applied to reduce overfitting. After the subsampling process in the max-pooling layer, the output is of size $104 \times 1 \times 64$ and it is then flattened to a size of 6556×1 before two fully connected layers of size 128 and 45 are applied. The first FC layer is used to further reduce the dimensionality of output from the max-pooling layer and it is activated with ReLU. The second FC layer is activated with the softmax function (Goodfellow *et al.*, 2016) to output the probability of the input PDF being one of the 45 space groups considered in our study.

Categorical cross entropy loss (Bishop, 2006) is used for training our model. It is apparent from Table 1 that the number of data entries in each space group are not evenly distributed, varying from 373 ($I\bar{4}2d$) to 7392 ($P2_1/c$) per space group. We would like to avoid the possibility of obtaining a neural network that is biased towards space groups with abundant data entries. To mitigate the effect of the unbalanced data set, loss from each training sample is multiplied by a class weight (King & Zeng, 2001) which is the inverse of the ratio between the number of data entries from the same space-group label in the training sample and the size of the entire training set. We then use adaptive moment estimation (Adam) (Kingma & Ba, 2014) as the stochastic optimization method to train our model with a mini-batch size of 64. During the training step, we follow the same protocol outlined in the work of He *et al.* (2016) to perform the weight initialization (He *et al.*, 2015) and batch normalization (Ioffe & Szegedy, 2015). A dropout strategy (Srivastava *et al.*, 2014) is also applied in the pooling layer to reduce overfitting in our neural network. The parameters in the CNN model are iteratively updated through the stochastic gradient descent method (Adam).

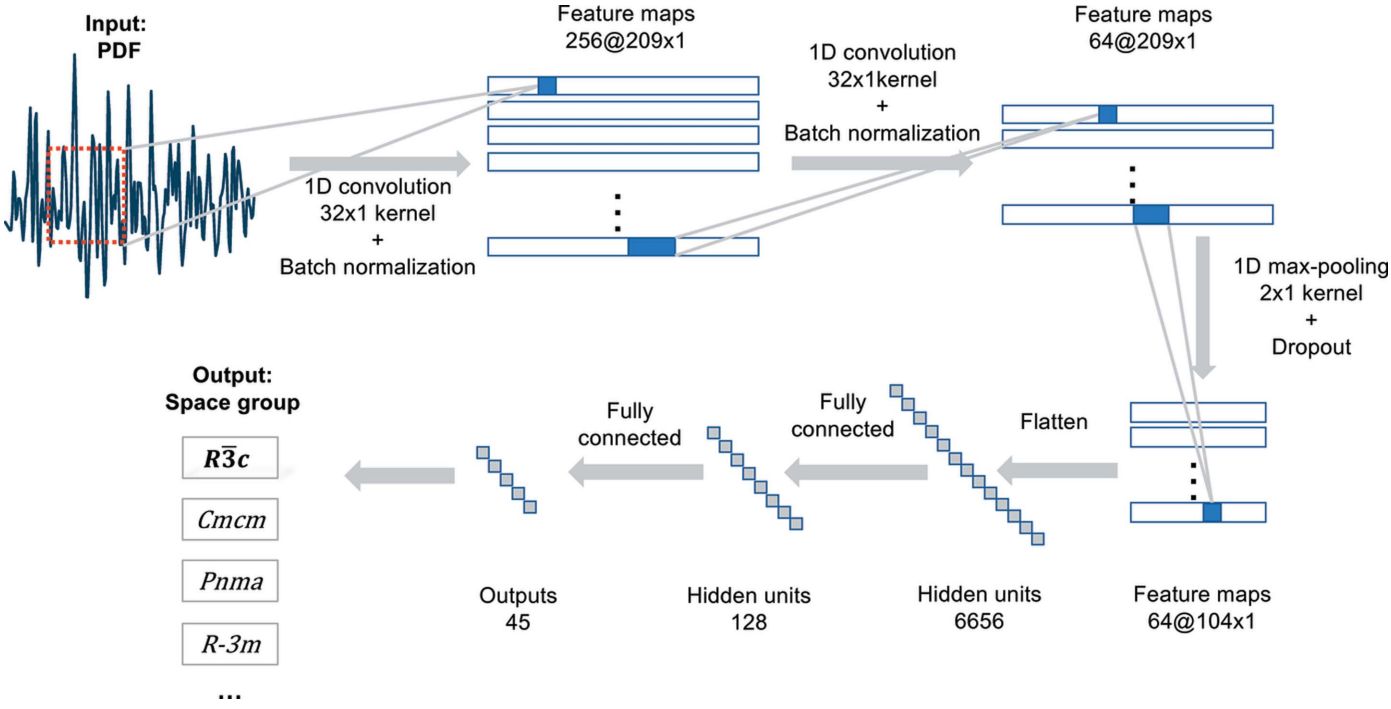


Figure 4
Schematic of our convolutional neural network (CNN) architecture.

Learning rate is a parameter that affects how drastically the parameters are updated at each iteration. A small learning rate is preferable when the parameters are close to some set of optimal values and *vice versa*. Therefore, an appropriate schedule of learning rate is crucial for training a model. Our training starts with a learning rate of 0.1, and the value is reduced by a factor of 10 at epochs 81 and 122. With the learning rate schedule described, the optimization loss against the testing set, along with the prediction accuracy on the training and testing sets, are plotted with respect to the number of epochs in Fig. 5. Our training is terminated after 164 epochs when the training accuracy, testing accuracy and optimization loss all plateau, meaning no significant improvement to the model would be gained with further updates to the parameters.

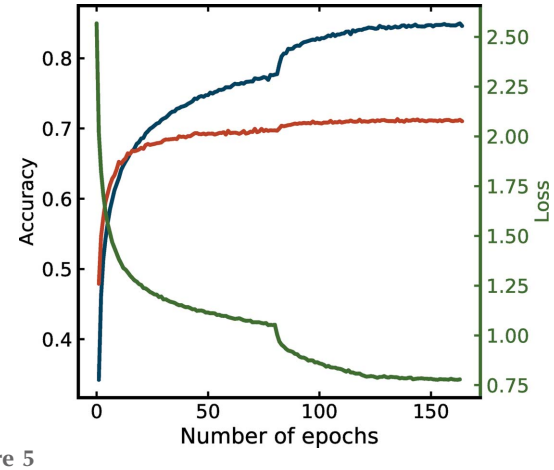


Figure 5
Accuracy of the CNN model on the training set (blue), the testing set (red) and the optimization loss against the testing set (green) with respect to number of epochs during the training step.

Our CNN model is implemented with *Keras* (Chollet *et al.*, 2015) and trained on a single Nvidia Tesla K80 GPU.

Under the architecture and training protocol discussed above, our best CNN model yields an accuracy of 70.0% from top-1 prediction and 91.9% from top-6 predictions, which outperforms the LR model by 15%. Similarly, from Fig. 2, we observe a more than 10% improvement in the alternative accuracy after considering top-2 predictions (ΔA_2) in the CNN model and the improvement (ΔA_i) decreases monotonically, even on a more drastic trend than the case of the LR model, when more predictions are considered.

4. Results and discussion

4.1. Space-group determination on calculated PDFs

The main result of the work is that, for the CNN model and defining success that the correct space group is found in the top-6 choices, we achieve a greater than 90% success rate (the correct space group is returned in the top position 70% of the time) when just the normalized PDF is given to the ML model. This success rate is much greater than random guessing and suggests that this approach may be a practically useful way of getting space-group information from PDFs. Below we explore in greater detail the performance of the CNN, including analyzing how it fails when it gets the answer wrong.

In general, it is fair to expect an ML model to achieve a higher accuracy on a space group with abundant training

samples. However, from Fig. 3, it is clear that the LR model even fails to identify well-represented space groups across all space-group numbers. On the other hand, a positive correlation between the size of the training data and the classification ratio is observed in the CNN model. Furthermore, except for space group $Ia\bar{3}d$, which is the most symmetric space group, the classification ratios on the rarely seen groups are lower than the well-represented groups in our CNN model. However, the main result is that the CNN performs significantly better than the LR model for all space groups, especially on structures with lower symmetry. There is an overall trend towards increase in the prediction ability as the symmetry increases, and there are outliers, but there seems to be a trend that the CNN model is better at predicting space groups for more highly populated space groups.

The confusion matrix (Stehman, 1997) is a common tool to assess the performance of an ML model. The confusion matrix, **M**, is an N-by-N matrix, where N is the number of labels in the data set. The rows of **M** identify the true label (correct answer) and the columns of **M** mean the label predicted by the model. The numbers in the matrix are the proportion of results in each category. For example, the

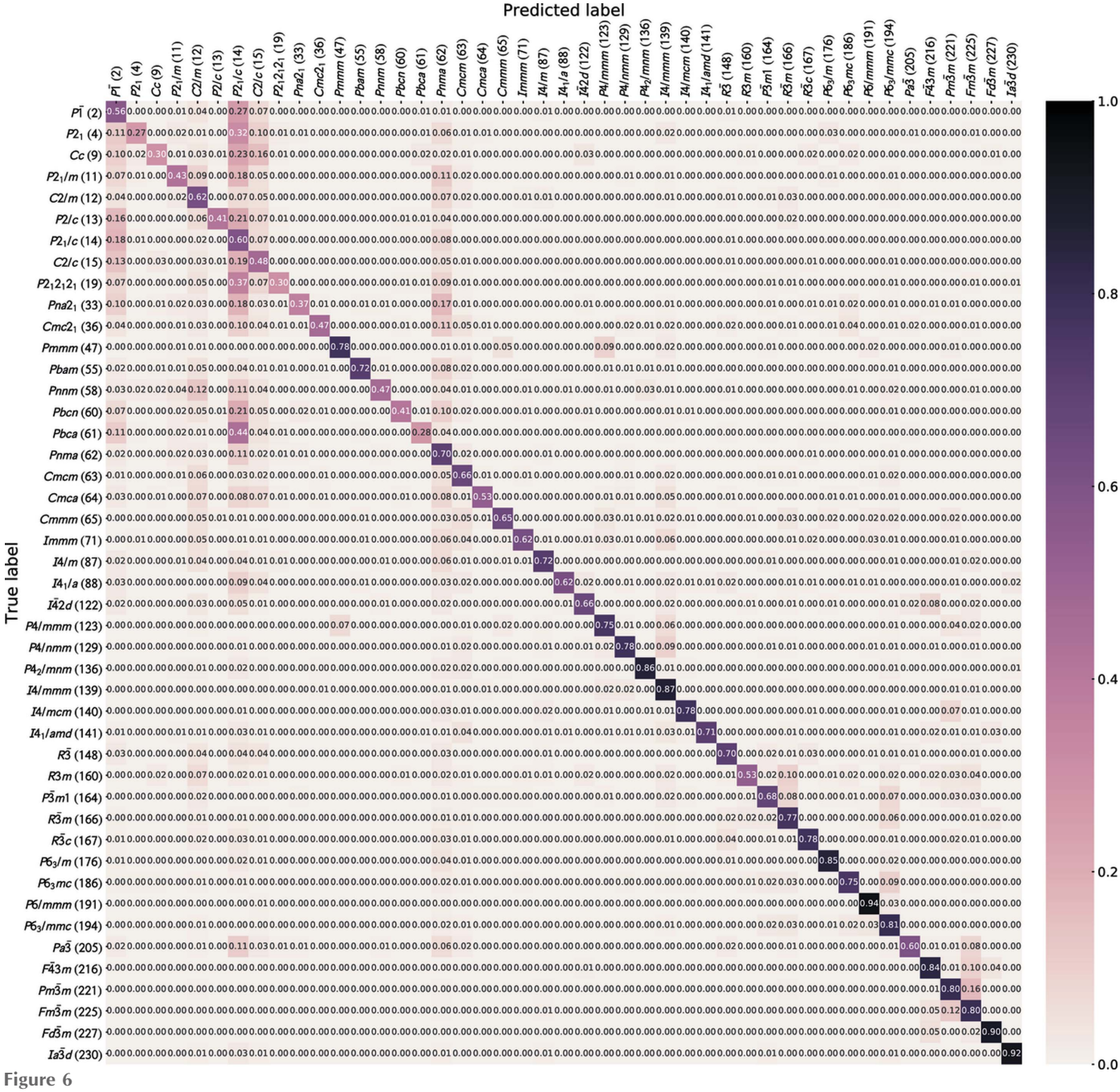


Figure 6
The confusion matrix of our CNN model. The row labels indicate the correct space group and the column labels the space group returned by the model. An ideal model would result in a confusion matrix with all diagonal values being 1 and all off-diagonal values being zero. The numbers in parentheses are the space-group number.

Table 3
Top-6 space-group predictions from the CNN model on experimental PDFs.

Entries in **bold** are the most probable space group from existing literature listed in the References column. More than one prediction are highlighted when these space groups are regarded as highly similar in the literature. Details about these cases are discussed in the text. The Note column specifies if the PDF is from a crystalline (C) or nanocrystalline (NC) sample. The experimental data were collected under various instrumental conditions which are not identical to the training set and experimental data were measured at room temperature, unless otherwise specified.

Sample	1st	2nd	3rd	4th	5th	6th	References	Note
Ni	***Fm$\bar{3}$m***	*Pm$\bar{3}$m*	*Fd$\bar{3}$m*	*F$\bar{4}$3m*	*P4/mmm*	*P6$_3$/mmc*	Owen & Yates (1936)	C
Fe_3O_4	***Fd$\bar{3}$m***	*I4$_1$/amd*	*R$\bar{3}$m*	*Fm$\bar{3}$m*	*F$\bar{4}$3m*	*P6$_3$/mmc*	Fleet (1981)	C
CeO_2	***Fm$\bar{3}$m***	*Fd$\bar{3}$m*	*Pm$\bar{3}$m*	*F$\bar{4}$3m*	*Pa$\bar{3}$*	*P4/mmm*	Yashima & Kobayashi (2004)	C
Sr_2IrO_4†	*Fm$\bar{3}$m*	*P6/mmm*	*P6$_3$/mmc*	*Pm$\bar{3}$m*	*Fd$\bar{3}$m*	*R$\bar{3}$m*	Huang *et al.* (1994), Shimura *et al.* (1995)	C
$CuIr_2S_4$	***Fd$\bar{3}$m***	*Fm$\bar{3}$m*	*F$\bar{4}$3m*	*R$\bar{3}$m*	*Pm$\bar{3}$m*	*R3m*	Furubayashi *et al.* (1994)	C
CdSe†	*P2$_1$/c*	*P$\bar{1}$*	*C2/c*	*Pnma*	*Pna2$_1$*	*P2$_1$2$_1$2$_1$*	Masadeh *et al.* (2007)	C
$IrTe_2$	*C2/m*	***P$\bar{3}$m1***	*P2$_1$/c*	*P$\bar{1}$*	*P2$_1$/m*	*C2/c*	Matsumoto *et al.* (1999), Yu *et al.* (2018)	C
$IrTe_2$@10 K	***C2/m***	*P6$_3$/mmc*	*P6/mmm*	*P4/mmm*	***P$\bar{1}$***	*P2$_1$/c*	Matsumoto *et al.* (1999), Toriyama *et al.* (2014)	C
Ti_4O_7	***P$\bar{1}$***	*C2/c*	*P2$_1$/c*	*C2/m*	*Pnnm*	*P4$_2$/mnm*	Marezio & Dernier (1971)	C
$MAPbI_3$@130 K	*P$\bar{1}$*	*P2$_1$/c*	*C2/c*	*P2$_1$2$_1$2$_1$*	***Pnma***	*Pna2$_1$*	Swainson *et al.* (2003)	C
$MoSe_2$	***P6$_3$/mmc***	*R3m*	*R$\bar{3}$m*	*P6$_3$mc*	*P4/mmm*	*Fd$\bar{3}$m*	James & Lavik (1963)	C
TiO_2 (anatase)	***I4$_1$/amd***	*C2/m*	*P2$_1$/m*	*C2/c*	*P$\bar{1}$*	*P2$_1$/c*	Horn *et al.* (1972)	NC
TiO_2 (rutile)	***P4$_2$/mnm***	*C2/m*	*P2$_1$/c*	*P$\bar{1}$*	*P2$_1$/m*	*Pnma*	Baur & Khan (1971)	NC
Si†	*P6$_3$mc*	*I$\bar{4}$2d*	*R3m*	*C2/c*	*P$\bar{1}$*	*Pbca*	Rohani *et al.* (2019)	NC
$BaTiO_3$	***R3m***	***P4/mmm***	*C2/m*	*P6$_3$/mmc*	*Pnma*	*Cmcm*	Kwei *et al.* (1993), Page *et al.* (2010)	NC

† Indicates where the CNN model fails to predict the correct space group.

diagonal elements indicate the proportion of outcomes where the correct label was predicted in each case, and the matrix element in the *Fd$\bar{3}$m* row and the *F$\bar{4}$3m* column (value 0.05) is the proportion of PDFs from an *Fd$\bar{3}$m* space-group structure that were incorrectly classified as being in space group *F$\bar{4}$3m*. For an ideal prediction model, the diagonal elements of the confusion matrix should be 1.0 and all off-diagonal elements would be zero. The confusion matrix from our CNN model is shown Fig. 6.

We observe 'teardrop' patterns in the columns of *P$\bar{1}$*, *P2$_1$/c* and *Pnma*, meaning the CNN model tends to incorrectly assign a wide range of space groups into these groups. On the surface, this behavior is worrying but the confusions actually correspond to the real group–subgroup relation which has been known and tabulated in the literature (Ascher *et al.*, 1969; Boyle & Lawrenson, 1972; Hahn, 2002). For the case of *P$\bar{1}$*, the major confusion groups (*P2$_1$/c*, *C2/c* and *P2/c*) are in fact minimal non-isomorphic supergroups of *P$\bar{1}$*. Moreover, *P2$_1$2$_1$2$_1$* shares the same subgroup (*P2$_1$*) with *P2$_1$/c* and *Pbca* is a supergroup of *P2$_1$2$_1$2$_1$* while *Pbcn* is a supergroup of *P2$_1$/c*. Similar reasoning can be applied to the case of *P2$_1$/c* and *Pnma* as well. The statistical model appears to be picking up some real underlying mathematical relationships.

We also investigate the cases with low classification accuracy (low value in diagonal elements) from the CNN model. *P2$_1$* is the group with the lowest accuracy (27%) among all labels. The similar group–subgroup reasoning holds for this case as well. *P2$_1$/c* (32% error rate) is, again, a supergroup of *P2$_1$* and *C2/c* (10% error rate) is a supergroup of *P2$_1$/c*. The same reasoning holds for other confusion cases and we will not explicitly go through it here, but this suggests that these closely group/subgroup-related space groups should also be considered whenever the CNN model returns another one in the series. It is possible to train a different CNN model which focuses on disambiguating space groups that are closely related by the group/subgroup relationship. However, we did not implement this kind of hierarchical model in our study.

4.2. Space-group determination on experimental PDFs

The CNN model is used to determine the space group of 15 experimental PDFs and the results are reported in Table 3. For each experimental PDF, structures are known from previous studies which are also referenced in the table. Both crystalline (C) and nanocrystalline (NC) samples with a wide range of structural symmetries are covered in this set of experimental PDFs. It is worth noting that the sizes of the NC samples chosen are roughly equal to or larger than 10 nm, at which size, in our measurements, the PDF signal from the NC material falls off roughly at the same rate as that from crystalline PDFs in the training set. Every experimental PDF is subject to experimental noise and collected under various instrumental conditions that result in aberrations to the PDF that are not identical to parameter values used to generate our training set (Table 2). It is therefore expected that the CNN classifier will work less well than on the testing set. From Table 3, it is clear that the CNN model yields an overall satisfactory result in determining space groups from experimental data with the space group from 12 out of 15 test cases properly identified in the top-6 predictions.

Here we comment on the performance of the CNN. In the cases of $IrTe_2$ at 10 K, the material has been reported in the literature in both *C2/m* and *P$\bar{1}$* space groups (Matsumoto *et al.*, 1999; Toriyama *et al.*, 2014), and it is not clear which is correct. The CNN returned both space groups in the top-6. Furthermore, for data from the same sample at room temperature, the CNN model identifies not only the correct space group (*P$\bar{3}$m1*), but also the space groups that the structure will occupy below the low-temperature symmetry-lowering transition (*C2/m*, *P$\bar{1}$*). For the case of $BaTiO_3$

nanoparticles, the CNN model identifies two space groups that are considered in the literature to yield rather equivalent explanatory power (*R3m*, *P*4/*mmm*) (Kwei *et al.*, 1993; Page *et al.*, 2010). It is encouraging that the CNN appears to be getting the physics right in these cases.

Investigating the failing cases from the CNN model (entries with a dagger in Table 3) also reveals insights into the decision rules learned by the model. Sr_2IrO_4 was firstly identified as a perovskite structure with space group *I*4/*mmm* (Randall *et al.*, 1957), but later work pointed out that a lower-symmetry group $I4_1/acd$ is more appropriate due to correlated rotations of the corner-shared IrO_6 octahedra about the *c* axis (Huang *et al.*, 1994; Shimura *et al.*, 1995). There is a long-wavelength modulation of the rotations along the *c* axis resulting in a supercell with a five-times expansion along that direction (a = 5.496, c = 25.793 Å). The PDF will not be sensitive to such a long-wavelength superlattice modulation which may explain why the model does not identify a space group close to the $I4_1/acd$ space group, reflecting additional symmetry breaking due to the supermodulation. It is not completely clear what the space group would be for the rotated octahedra without the supermodulation, so we are not sure if this space group is among the top-6 that the model found.

Somewhat surprisingly the CNN fails to find the right space group for wurtzite CdSe, which is a very simple structure, but rather finds space groups with low symmetries. One possible reason is that we know there is a high degree of stacking faulting in the bulk CdSe sample that was measured. This was best modeled as a phase mixture of wurtzite (space group $P6_3mc$) and zinc-blende (space group $F\overline{4}3m$) (Masadeh *et al.*, 2007). The prediction of low-symmetry groups might reflect the fact the underlying structure cannot be described with a single space group.

5. Conclusion

We demonstrate an application of machine learning (ML) to determine the space group directly from an atomic pair distribution function (PDF). We also present a convolutional neural network (CNN) model which yields a promising accuracy (91.9%) from the top-6 predictions when it is evaluated against the testing data. Interestingly, the trained CNN model appears to capture decision rules that agree with the mathematical (group–subgroup) relationships between space groups. The trained CNN model is tested against 15 experimental PDFs, including crystalline and nanocrystalline samples. Space groups from 12 of these experimental data sets were successfully found in the top-6 predictions by the CNN model. This shows great promise for preliminary, model-independent assessment of PDF data from well-ordered crystalline or nanocrystalline materials.

APPENDIX *A*
Logistic regression and elastic net regularizations

Consider a data set with a total M structures and K distinct space-group labels. Each structure has a space group and we denote the space group of the mth structure as k_m where $k_m \in \{1, 2, \ldots K\}$, our complete set of space groups. In the setup of the LR model, the probability of a feature x_m of dimension d, which is a computable from the mth structure, belonging to a specific space group k_m is parametrized as

$$\Pr(k_m|x_m, \beta^{k_m}) = \frac{\exp\left(\beta_0^{k_m} + \sum_{i=1}^{d} \beta_i^{k_m} x_{m,i}\right)}{1 + \exp\left(\beta_0^{k_m} + \sum_{i=1}^{d} \beta_i^{k_m} x_{m,i}\right)}, \quad (6)$$

where $\beta^{k_m} = \{\beta_0^{k_m}, \beta_1^{k_m}, \ldots, \beta_d^{k_m}\}$ is a set of parameters to be determined. The index k_m runs from 1 to 45 which corresponds to the total number of space groups considered in our study. Since the space group k and feature x are both known for the training data, the learning algorithm is then used to find an optimized set of $\beta = \{\beta^{k_m} : k_m = 1, 2, \ldots, K\}$ which maximizes the overall probability of determining the correct space group $\Pr(k_m|x_m, \beta^{k_m})$ on all M training data.

For each of the M structures, there will be a binary result for classification: either the space-group label is correctly classified or not. This process can be regarded as M independent Bernoulli trials. The probability function for a single Bernoulli trial is expressed as

$$f(k_m|x_m, \boldsymbol{\beta}^{k_m}) = \left[\Pr(k_m|x_m, \boldsymbol{\beta}^{k_m})\right]^{\gamma_m} \left[1 - \Pr(k_m|x_m, \boldsymbol{\beta}^{k_m})\right]^{1-\gamma_m}, \quad (7)$$

where γ is an indicator. $\gamma_m = 1$ if the space-group label k_m is correctly predicted and $\gamma_m = 0$ if the prediction is wrong. Since each classification is independent, the joint probability function for M classifications on the space-group label, $f_M(\mathbf{K}|x, \boldsymbol{\beta})$, is written as

$$f_M(\mathbf{K}|\mathbf{x}, \boldsymbol{\beta}) = \prod_{m=1}^{M} f(k_m|x_m, \boldsymbol{\beta}^{k_m}), \quad (8)$$

where $\mathbf{K} = \{k_m\}$ and $\mathbf{x} = \{x_m\}$. Furthermore, since both the label and features are known in the training set, equation (8) is just a function of β,

$$L(\beta) = f_M(\mathbf{K}|\mathbf{x}, \boldsymbol{\beta}). \quad (9)$$

Logarithm is a monotonic transformation. Taking the logarithm of equation (9) does not change the original behavior of the function and it improves the numerical stability as the product of probabilities is turned into the sum of the logarithm of probabilities and extreme values from the product can still be computed numerically. We therefore arrive at the 'log-likelihood' function:

$$l(\beta) = \log[L(\beta)]. \quad (10)$$

It is common to include 'regularization' (Hastie *et al.*, 2009) for reducing overfitting in the model. The regularization scheme chosen in our implementation is 'elastic net' which is known for encouraging sparse selections on strongly correlated variables (Zou & Hastie, 2005). The explicit definitions of the log-likelihood function with elastic regularization are written as

$$l_t(\beta) = l(\beta) + \alpha\left[\Lambda\|\beta\|_1 + (1 - \Lambda)\|\beta\|_2^2\right], \quad (11)$$

Table 4
Accuracies of the CNN model with different sets of hyperparameters.

The last row specifies the optimum set of hyperparameters for our final CNN model.

No. filters	Kernel size	No. hidden units	No. ensembles	Top-1 accuracy (%)	Top-6 accuracy (%)
128, 32	24	128	2	64.1	90.7
256, 64	24	128	2	68.6	91.6
64, 64	24	128	2	67.4	91.1
128, 64	32	128	2	69.0	91.7
128, 64	16	128	2	66.6	91.3
128, 64	24	256	2	69.2	91.6
128, 64	24	64	2	66.4	91.2
128, 64	24	128	1	65.7	91.1
128, 64	24	128	3	68.2	91.6
256, 64	32	128	3	70.0	91.9

where $\|\cdot\|$ and $\|\cdot\|_2^2$ stand for the L1 and L2 norm (Horn, 2012), respectively. Two hyperparameters α and Λ are introduced under this regularization scheme. α is a hyperparameter that determines the overall 'strength' of the regularization and Λ governs the relative ratio between L1 and L2 regularization (Zou & Hastie, 2005). Describing the detailed steps in optimizing equation (11) is beyond the scope of this paper, but they are available in most of the standard ML reviews (Hastie *et al.*, 2009; Bishop, 2006).

APPENDIX *B*
Robustness of the CNN model

The classification accuracies from CNN models with different sets of hyperparameters, such as number of filters, kernel size and pooling size, are reproduced in Table 4. The classification accuracy only varies modestly across different sets of hyperparameters and this implies the robustness of our CNN architecture. We determined the desired architecture of our CNN model based on the classification accuracy on the testing set and the learning curves (loss, training accuracy and testing accuracy) reported in Fig. 5.

Funding information

Funding for this research was provided by: National Science Foundation, Division of Materials Research (grant No. 1534910); National Science Foundation, Division of Mathematical Sciences (grant No. 1719699); National Science Foundation, Division of Computing and Communication Foundations (grant No. 1704833). X-ray PDF measurements were conducted on beamline 28-ID-1 (PDF) and 28-ID-2 (XPD) of the National Synchrotron Light Source II, a US Department of Energy (DOE) Office of Science User Facility operated for the DOE Office of Science by Brookhaven National Laboratory under Contract No. DE-SC0012704.

References

Altomare, A., Camalli, M., Cuocci, C., Giacovazzo, C., Moliterni, A. & Rizzi, R. (2009). *J. Appl. Cryst.* **42**, 1197–1202.

Altomare, A., Campi, G., Cuocci, C., Eriksson, L., Giacovazzo, C., Moliterni, A., Rizzi, R. & Werner, P.-E. (2009). *J. Appl. Cryst.* **42**, 768–775.

Ascher, E., Gramlich, V. & Wondratschek, H. (1969). *Acta Cryst.* B**25**, 2154–2156.

Bahdanau, D., Cho, K. & Bengio, Y. (2014). arXiv:1409.0473 [cs.CL].

Baur, W. H. & Khan, A. A. (1971). *Acta Cryst.* B**27**, 2133–2139.

Belsky, A., Hellenbrandt, M., Karen, V. L. & Luksch, P. (2002). *Acta Cryst.* B**58**, 364–369.

Billinge, S. J. L., Duxbury, P. M., Gonalves, D. S., Lavor, C. & Mucherino, A. (2018). *Ann. Oper. Res.* pp. 1–43.

Billinge, S. J. L. & Levin, I. (2007). *Science*, **316**, 561–565.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. New York: Springer-Verlag, Inc.

Boultif, A. & Louër, D. (2004). *J. Appl. Cryst.* **37**, 724–731.

Boyle, L. L. & Lawrenson, J. E. (1972). *Acta Cryst.* A**28**, 489–493.

Choi, J. J., Yang, X., Norman, Z. M., Billinge, S. J. L. & Owen, J. S. (2014). *Nano Lett.* **14**, 127–133.

Chollet, F., *et al.* (2015). *Keras*. https://keras.io.

Cliffe, M. J., Dove, M. T., Drabold, D. A. & Goodwin, A. L. (2010). *Phys. Rev. Lett.* **104**, 125501.

Coelho, A. A. (2003). *J. Appl. Cryst.* **36**, 86–95.

Coelho, A. A. (2017). *J. Appl. Cryst.* **50**, 1323–1330.

Dahl, G. E., Sainath, T. N. & Hinton, G. E. (2013). *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 8609–8613.

Egami, T. & Billinge, S. J. L. (2012). *Underneath the Bragg Peaks: Structural Analysis of Complex Materials*, 2nd ed. Amsterdam: Elsevier.

Farrow, C. L. & Billinge, S. J. L. (2009). *Acta Cryst.* A**65**, 232–239.

Farrow, C. L., Juhás, P., Liu, J., Bryndin, D., Božin, E. S., Bloch, J., Proffen, T. & Billinge, S. J. L. (2007). *J. Phys. Condens. Matter*, **19**, 335219.

Fleet, M. E. (1981). *Acta Cryst.* B**37**, 917–920.

Furubayashi, T., Matsumoto, T., Hagino, T. & Nagata, S. (1994). *J. Phys. Soc. Jpn*, **63**, 3333–3339.

Giacovazzo, C. (1999). *Direct Phasing in Crystallography: Fundamentals and Applications*, 1st ed. Oxford University Press/ International Union of Crystallography.

Goodfellow, I., Bengio, Y. & Courville, A. (2016). *Deep Learning*. MIT Press.

Hahn, T. (2002). *International Tables for Crystallography*, Vol. A: *Space-group Symmetry*, 5th ed. Dordrecht: Springer.

Hastie, T., Tibshirani, R. & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. *Springer Series in Statistics*. New York: Springer-Verlag.

He, K., Zhang, X., Ren, S. & Sun, J. (2015). *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1026–1034.

He, K., Zhang, X., Ren, S. & Sun, J. (2016). *Computer Vision – ECCV 2016*, edited by B. Leibe, J. Matas, N. Sebe & M. Welling, *Lecture Notes in Computer Science*, pp. 630–645. New York: Springer International Publishing.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N. & Kingsbury, B. (2012). *IEEE Signal Process. Mag.* **29**, 82–97.

Horn, M., Schwerdtfeger, C. F. & Meagher, E. P. (1972). *Z. Kristallogr.* **136**, 273–281.

Horn, R. A. (2012). *Matrix Analysis*, 2nd ed. New York: Cambridge University Press.

Huang, Q., Soubeyroux, J. L., Chmaissem, O., Sora, I. N., Santoro, A., Cava, R. J., Krajewski, J. J. & Peck, W. F. (1994). *J. Solid State Chem.* **112**, 355–361.

Ioffe, S. & Szegedy, C. (2015). arXiv:1502.03167 [cs.LG].

James, G., Witten, D., Hastie, T. & Tibshirani, R. (2013). *An Introduction to Statistical Learning*, Vol. 103 of *Springer Texts in Statistics*. New York: Springer New York.

James, P. B. & Lavik, M. T. (1963). *Acta Cryst.* **16**, 1183.

Jarrett, K., Kavukcuoglu, K., Ranzato, M. & LeCun, Y. (2009). *2009 IEEE 12th International Conference on Computer Vision*, pp. 2146–2153.
Juhás, P., Cherba, D. M., Duxbury, P. M., Punch, W. F. & Billinge, S. J. L. (2006). *Nature*, **440**, 655–658.
Juhás, P., Granlund, L., Gujarathi, S. R., Duxbury, P. M. & Billinge, S. J. L. (2010). *J. Appl. Cryst.* **43**, 623–629.
Keen, D. A. & Goodwin, A. L. (2015). *Nature*, **521**, 303–309.
Kim, Y. (2014). arXiv:1408.5882 [cs.CL].
King, G. & Zeng, L. (2001). *Polit. Anal.* **9**, 137–163.
Kingma, D. P. & Ba, J. (2014). arXiv:1412.6980 [cs.LG].
Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012). *Advances in Neural Information Processing Systems 25*, edited by F. Pereira, C. J. C. Burges, L. Bottou & K. Q. Weinberger, pp. 1097–1105. Red Hook, New York, USA: Curran Associates, Inc.
Kwei, G. H., Lawson, A. C., Billinge, S. J. L. & Cheong, S.-W. (1993). *J. Phys. Chem.* **97**, 2368–2377.
LeCun, Y., Bengio, Y. & Hinton, G. (2015). *Nature*, **521**, 436–444.
Lecun, Y., Bottou, L., Bengio, Y. & Haffner, P. (1998). *Proc. IEEE*, **86**, 2278–2324.
Marezio, M. & Dernier, P. D. (1971). *J. Solid State Chem.* **3**, 340–348.
Markvardsen, A. J., Shankland, K., David, W. I. F., Johnston, J. C., Ibberson, R. M., Tucker, M., Nowell, H. & Griffin, T. (2008). *J. Appl. Cryst.* **41**, 1177–1181.
Masadeh, A. S., Božin, E. S., Farrow, C. L., Paglia, G., Juhás, P., Billinge, S. J. L., Karkamkar, A. & Kanatzidis, M. G. (2007). *Phys. Rev. B*, **76**, 115413.
Matsumoto, N., Taniguchi, K., Endoh, R., Takano, H. & Nagata, S. (1999). *J. Low Temp. Phys.* **117**, 1129–1133.
Mighell, A. D. & Santoro, A. (1975). *J. Appl. Cryst.* **8**, 372–374.
Neumann, M. A. (2003). *J. Appl. Cryst.* **36**, 356–365.
Owen, E. & Yates, E. (1936). *London Edinb. Dubl. Philos. Mag. J. Sci.* **21**, 809–819.
Page, K., Proffen, T., Niederberger, M. & Seshadri, R. (2010). *Chem. Mater.* **22**, 4386–4391.
Park, W. B., Chung, J., Jung, J., Sohn, K., Singh, S. P., Pyo, M., Shin, N. & Sohn, K.-S. (2017). *IUCrJ*, **4**, 486–494.
Pecharsky, V. K. & Zavalij, P. Y. (2005). *Fundamentals of Powder Diffraction and Structural Characterization of Materials.* New York, USA: Springer.
Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, E. (2011). *J. Mach. Learn. Res.* **12**, 2825.
Peterson, P. F., Božin, E. S., Proffen, Th. & Billinge, S. J. L. (2003). *J. Appl. Cryst.* **36**, 53–64.
Proffen, T., Page, K. L., McLain, S. E., Clausen, B., Darling, T. W., TenCate, J. A., Lee, S.-Y. & Ustundag, E. (2005). *Z. Kristallogr.* **220**, 1002–1008.
Radford, A., Metz, L. & Chintala, S. (2015). arXiv:1511.06434 [cs.LG].
Ramprasad, R., Batra, R., Pilania, G., Mannodi-Kanakkithodi, A. & Kim, C. (2017). *NPJ Comput. Mater.* **3**, 54.
Randall, J. J., Katz, L. & Ward, R. (1957). *J. Am. Chem. Soc.* **79**, 266–267.
Rohani, P., Banerjee, S., Ashrafi-Asl, S., Malekzadeh, M., Shahbazian-Yassar, R., Billinge, S. J. L. & Swihart, M. T. (2019). *Adv. Funct. Mater.* **29**, 1807788.
Shimura, T., Inaguma, Y., Nakamura, T., Itoh, M. & Morii, Y. (1995). *Phys. Rev. B*, **52**, 9143–9146.
Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T. & Hassabis, D. (2017). *Nature*, **550**, 354–359.
Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. (2014). *J. Mach. Learn. Res.* **15**, 1929–1958.
Stehman, S. V. (1997). *Remote Sens. Environ.* **62**, 77–89.
Sutskever, I., Vinyals, O. & Le, Q. V. (2014). *Advances in Neural Information Processing Systems 27*, edited by Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence & K. Q. Weinberger, pp. 3104–3112. Red Hook, New York, USA: Curran Associates, Inc.
Swainson, I. P., Hammond, R. P., Soullière, C., Knop, O. & Massa, W. (2003). *J. Solid State Chem.* **176**, 97–104.
Toriyama, T., Kobori, M., Konishi, T., Ohta, Y., Sugimoto, K., Kim, J., Fujiwara, A., Pyon, S., Kudo, K. & Nohara, M. (2014). *J. Phys. Soc. Jpn*, **83**, 033701.
Urusov, V. S. & Nadezhina, T. N. (2009). *J. Struct. Chem.* **50**, 22–37.
Visser, J. W. (1969). *J. Appl. Cryst.* **2**, 89–95.
Wolff, P. M. de (1957). *Acta Cryst.* **10**, 590–595.
Yashima, M. & Kobayashi, S. (2004). *Appl. Phys. Lett.* **84**, 526–528.
Yu, R., Banerjee, S., Lei, H. C., Sinclair, R., Abeykoon, M., Zhou, H. D., Petrovic, C., Guguchia, Z. & Bozin, E. (2018). *Phys. Rev. B*, **97**, 174515.
Ziletti, A., Kumar, D., Scheffler, M. & Ghiringhelli, L. M. (2018). *Nat. Commun.* **9**, 2775.
Zobel, M., Neder, R. B. & Kimber, S. A. J. (2015). *Science*, **347**, 292–294.
Zou, H. & Hastie, T. (2005). *J. R. Stat. Soc. Ser. B Stat. Methodol.* **67**, 301–320.