DeepStruc: Towards structure solution from pair distribution function data using deep generative models

Anonymous Author(s)
Affiliation
Address
email

Abstract

Structure solution of nanostructured materials that have limited long-range order remains a bottleneck in materials development. We present a deep learning algorithm, DeepStruc, that can solve a simple nanoparticle structure directly from a Pair Distribution Function (PDF) obtained from total scattering data by using a conditional variational autoencoder. We first apply DeepStruc to PDFs from seven different structure types of monometallic nanoparticles, and show that structures can be solved from both simulated and experimental PDFs, including PDFs from nanoparticles that are not present in the training distribution. We also apply DeepStruc to a system of *hcp*, *fcc* and stacking faulted nanoparticles, where DeepStruc recognizes stacking faulted nanoparticles as an interpolation between *hcp* and *fcc* nanoparticles and is able to solve stacking faulted structures from PDFs. Our findings suggests that DeepStruc is a step towards a general approach for structure solution of nanomaterials.

1 Introduction

Crystallographic methods, such as single crystal and powder diffraction, have been foundational in the development of functional materials over the past century. They yield atomic-scale structural models for crystalline materials and allow establishing the links between material structure and properties that are at the heart of materials development.[1,2] However, other approaches for structure determination are needed for nanostructured materials that have limited long-range order, and total scattering methods such as atomic pair distribution function (PDF) analysis have become increasingly important tools.[3-7] Currently, PDF analysis is mainly done by fitting a known starting model to an experimental PDF, a process known as structure refinement. Recent developments in automated modelling[8-10] have made it possible to extend the searched structural space, but identifying a model or solving a structure *de novo* from a PDF is still an enormous challenge. So far, only highly symmetrical nanostructures such as the C_{60} buckyball have been solved *ab initio* from a PDF.[11-15] Determining the structure of less symmetrical nanostructures is limited by the lost information caused by PDF peak overlap, which challenges the use of PDF for structure solution of more complicated nanomaterials.

An approach to handle the challenges due to the information barrier in PDFs is to employ supervised machine learning (ML) methods that can learn from well-known PDF-structure pairs. While determining a unique structure from a PDF is not always a solvable problem, as several different structures may give rise to identical PDFs, ML methods can still learn to capture the relationship between PDF and structure and thereby push the boundaries of nanostructure solution from PDF. When there is not enough information in the PDF to provide a unique structure solution, ML methods may provide a distribution of starting models which can aid in further structure analysis. In this work, we use deep

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

generative models (DGMs). DGMs are a class of ML models that can estimate the underlying data distribution from a reasonably small set of training examples.[16] A well-known use case of DGMs is in the generation of synthetic 'deep-fake' images[17,18] based on large datasets of real images. We here train our DGM to identify new structure models by training on known chemical structures. The DGM learns the relation between PDF and atomic structure, which enables it to solve a structure, based on a PDF it has not seen before and its learned chemical knowledge.

We apply our DGM, which we refer to as 'DeepStruc', for structural analysis of a model system of monometallic nanoparticles (MMNPs) with seven different structure types (Fig. 1a) and demonstrate the method for both simulated and experimental PDFs. DeepStruc is generative, which means that it can be used to construct structures that are not in the training set, i.e., solve a structure from a PDF. We demonstrate this capability on a dataset of face-centered cubic (*fcc*), hexagonal closed packed (*hcp*) and stacking faulted structures, where DeepStruc can recognize the stacking faulted structures as an interpolation between *fcc* and *hcp* and construct new structural models based on a PDF.

2 Method

2.1 Simulated and experimental data

To simulate the nanoparticles used in the training process of DeepStruc, the Python library atomic simulation environment (ASE) was used.[19] The seven different structure types: *fcc*, *bcc*, *sc*, *hcp*, icosahedral, decahedral, and octahedral were constructed with the cluster module in ASE in the same manner as described by Banerjee et al.[9] and Anker & Kjær et al.[20] All MMNPs were generated in sizes ranging from 5 to 200 atoms. Each MMNP was then populated with different atoms hence changing the lattice spacing/bond distances in the MMNP. To ensure that there were no duplicate MMNPs within the dataset, all MMNPs were decomposed into a distance list of all atom-atom distances. The distance lists are a reduced format of the xyz representation as they are rotation- and translation-invariant in Euclidean space. All the distance-lists were sorted and duplicate structures with equivalent distance lists were removed. This yielded a total of 3742 unique MMNPs, see section A in the Supplementary Information for the distribution of the seven structure types. The xyz-coordinates will be the label that DeepStruc has to reconstruct. Nanoparticles with each of the seven structure types can be seen in Fig. 1b along with their simulated PDF, Fig. 1c. The PDF is described mathematically in section B in the Supplementary information. All the simulation parameters used can been seen in section C in the Supplementary Information.

To further investigate the latent space behaviour of DeepStruc, a more chemically simple and intuitive dataset was made of *fcc*, *hcp*, and stacking faulted structures. *Fcc* and *hcp* can be considered layered structures that are only differentiated by the repetition of layers within the structure. *Fcc* consists of a repeated ABCABC layered structure where *hcp* is an ABABAB layered structure. A 5 layered stacking fault structure could then be described as ABCAC, as it does not satisfy either of the *fcc* or *hcp* stacking criteria, see Fig. S8. A total of 1620 stacking fault structures were generated.

2.2 Data representation

In this work, the structures from ASE are converted into a graph-based representation in order to capture the interatomic relationships, as the original representation generated with ASE are not optimal as input to DeepStruc. Graph representations have seen increasing success in machine learning applications related to materials science as the interatomic relations in graphs are invariant to transformations of the structure such as solid translations and rotations.[21,22] Each structure in graph representation can be described as $G = (\mathbf{X}, \mathbf{A})$, where $\mathbf{X} \in \mathbb{R}^{N \times F}$ is the node feature matrix which contains F features that can describe each of the N atoms in the structure. We use $F = 3$ comprising only the Euclidean coordinates of the atom in a 3-dimensional space. The interatomic relationships are captured using the adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$. In our case, the entries of the adjacency matrix are the Euclidean distance between each pair of atoms, resulting in a soft adjacency matrix. However, to make the adjacency matrix sparse, when the distance between any pair of nodes is larger than the lattice constant the corresponding edge weight is set to zero. When the edge weight is zero this corresponds to absence of an edge between the pair of nodes, and in other cases the edges have a weight given by the interatomic distance. Section D in the Supplementary Information shows a decahedron consisting of seven atoms alongside the components describing it in our chosen graph representation.

2.3 The Conditional Deep Generative Model (DGM)

DGMs such as variational autoencoders (VAEs) are commonly used to synthesize novel, synthetic data by approximating the underlying data-generating processes based on the training data.[23] In this work, we are interested in generating structures based on properties such as the PDF resulting in the conditional DGM scenario. The specific formulation of the conditional DGM used in this work is the CVAE, initially proposed for computer vision tasks[24] and more recently it has also been explored for synthesizing novel drug molecules.[25] The CVAE in this work is trained to solve the unassigned distance geometry problem[26] (uDGP) as it solves the task of converting the distances within a PDF to a chemical structure. In the uDGP the problem of taking a starting point of a list of distances and reconstructing it into a structure is broken down into two discrete problems. First, is to discover the graph that connects pairs of atoms, with the edges labelled by the distances from the distance list (the assignment problem). Second is to embed this graph into Euclidean space. An illustration of the CVAE can be seen in Fig. 1a. Here, the blue area is the training process, and the green area is the prediction/inference process. During training of the CVAE, the encoder takes pairs of structures and their corresponding PDFs as input. The encoder learns to map the structure-PDF pairs into a low-dimensional, latent Gaussian distribution, known as the encoder distribution. Each structure-PDF pair is mapped to certain regions of the latent space. When trained with large amounts of diverse data, the latent space is able to capture relationships between different structures and PDF pairs so that similar structures are closer in this latent space than very different structures. CVAEs are different from classical autoencoders in that the latent space is probabilistic, which makes it possible to sample structures from these latent encoder distributions. This is achieved during training by forcing the encoder distributions to align with a simpler prior distribution which only takes the PDF as input. The two distributions are matched by minimizing the Kullback-Leibler Divergence between the encoder and prior distributions and is interpreted as the regularization term, L_{reg}.

The prior NN gets the PDF as input and maps it to the low-dimensional prior distribution. The low-dimensional latent vector conditioned on the PDF is then input to the decoder, which is tasked to predict the xyz-coordinates of the structural input. During the training process, the mean squared error (MSE) between the xyz-coordinates of the input and output are computed to force the decoder to predict xyx-coordinates from the latent representations. The MSE is defined as the reconstruction loss, L_{rec}. The CVAE is trained by jointly optimizing these two loss components:

$$L_{CVAE} = L_{rec} + \beta \cdot L_{reg}$$

Where β is a scaling factor that controls the relative influence of the regularization- and reconstruction-terms. In our training process, at initialization β is set to 0 which allows the model to focus on minimizing L_{rec}. Each time L_{rec} gets below a certain threshold β is increased. This helps keep the model from falling into a local minimum and the process is repeated until convergence has been reached. Similar strategies for annealing β in VAEs have been attempted.[27,28] At inference (test) time, the prior NN receives the PDF as input which is then mapped to the low-dimensional latent space which during training has been trained to match the encoder distribution. A sufficiently well trained CVAE is then able to predict structures from the latent space based on the PDF input. A simplified version of the CVAE used for this work, DeepStruc, can be seen in Fig. 1a. The CVAE is presented more formally in the work by Anker & Kjær et al..[20]

2.4 Graph Conditional Variational Autoencoder (CVAE)

In this work, two types of CVAEs were utilized depending on the type of encoder. In the conventional CVAE, the encoder was based on Multi-Layered Perceptrons which operate on a tabular format of the node features, and the adjacency matrix populated with atom–atom distances. For the second type of CVAE – that we call the graph CVAE – the encoder consists of a graph neural network (GNN)[22,29] and is able to process graph structured data, taking the neighbourhood information into consideration. GNNs are generalized message passing methods that can aggregate information from the neighbourhood of a node by passing messages along the edges. These messages are learned during training and can summarize the information present at the node necessary for the downstream tasks. Further, by making the encoder deep, i.e. adding additional GNN layers, nodes can get access to information from nodes that are farther from them. For instance, in a k-layered GNN each node had access to information from nodes that are k-hops away. Further, we were able to obtain comparable reconstruction quality from the graph CVAE with only two latent dimensions compared to using

eight dimensions for the conventional CVAE. This indicates that the graph encoder is able to better compress the information present in the node and adjacency matrices. A minor technical detail in our CVAE models is that the predictions from the decoder do not exactly match the input features. That is, the decoder does not reconstruct the full input comprising node features and adjacency matrix but only the node features. The algorithm we refer to as DeepStruc is a graph based CVAE.

3 Results

3.1 Training DeepStruc to determine the structure of MMNPs from PDF data

DeepStruc, illustrated in Fig. 1a and discussed below, is a graph-based conditional variational autoencoder (graph CVAE). Autoencoders are a class of deep learning (DL) methods where high-dimensional inputs, such as chemical structures,[20,25] are reduced in dimensionality. The transformation into 2 or 3 dimensional vectors is achieved using an information bottleneck by an encoder neural network (NN),[20,30,31] and the resulting lower-dimensional, compressed feature space is known as the latent space. A decoder NN can reconstruct the input from these low-dimensional representations. When the latent space is regularized (smoothed) using normal distributions instead of discrete points we obtain a variational autoencoder (VAE). We have previously demonstrated that VAEs does a better job interpolating in the latent space compared to deterministic AEs.[20] The VAE can be made to be dependent (conditioned) on additional information by the prior NN resulting in a CVAE.[31]

We here use MMNP structures (Fig. 1b) as input, and condition them on their simulated PDFs (Fig. 1c). The MMNP structures span seven different structure types computed using a variety of metals to emulate the variability in bond lengths in real metallic nanoparticle samples. The structure types are simple cubic (*sc*), body-centered cubic (*bcc*), face-centered cubic (*fcc*), hexagonal closed packed (*hcp*), decahedral, icosahedral, and octahedral, and all structure types have been constructed in sizes from 5 to 200 atoms. We used 3743 MMNP structures, which were randomly split into training- (60 %), validation- (20 %) and testing-sets (20 %). Note that the validation and test sets are derived from the same underlying data distribution as the training set, and serve as intermediaries to the actual test set which is based on the experimental PDF data. A histogram of the distribution of the seven structure types are provided in section A in the Supplementary Information. During the training process (blue + green region Fig. 1a), DeepStruc learns to map the conditioning PDFs to their structures in the latent space. After the training process is complete, DeepStruc can be used on data that have not been part of the training set, which is referred to as 'inference'.

3.2 Mapping of structures in a latent space

We first evaluate DeepStruc's ability to map the MMNP structures in a low-dimensional latent space by investigating structural trends and clustering. Fig. 2 shows a visualization of the two-dimensional latent space with selected MMNP reconstructions indicated. The colour of the points indicates the structure type, and the relative point size indicates the size of the MMNP cluster. We observe that DeepStruc learns to map the chemical structures in the latent space by size and symmetry. It maps the cubic structure types (*sc*, *bcc*, and *fcc*) together, and it learns that the octahedral MMNPs are closely related to the *fcc* structure type. Interestingly, DeepStruc also allocates the decahedral structures to be in between the *fcc* and *hcp* structures. This can be rationalized by considering that decahedral structures are constructed from five tetrahedrally shaped *fcc* crystals which are separated by $\{111\}$ twin boundaries that resemble stacking faults.[9,32,34] The twin boundaries will resemble stacking faulted regions of *fcc* justifying that they exist in the latent space between *fcc* and *hcp*.

3.3 DeepStruc for structure determination from PDF

We now move on to identify structures directly from a PDF. The results of using DeepStruc on seven simulated PDFs of MMNPs not used in the training process are illustrated in Fig 3. Here, we show the structure that the input PDF was calculated from (left), the reconstructed structure (right), and its agreement with the input PDF after structure refinement (middle, discussed below). In all seven cases, the structures are correctly reconstructed from the PDF input. Before structure refinement, the mean absolute error (MAE) of the atom positions is 0.128 ± 0.073 as described in section E in the Supplementary Information. However, the MAE is artificially high due to a common aberration by DeepStruc, where it predicts the right geometric atomic arrangement, but isotropically contracted

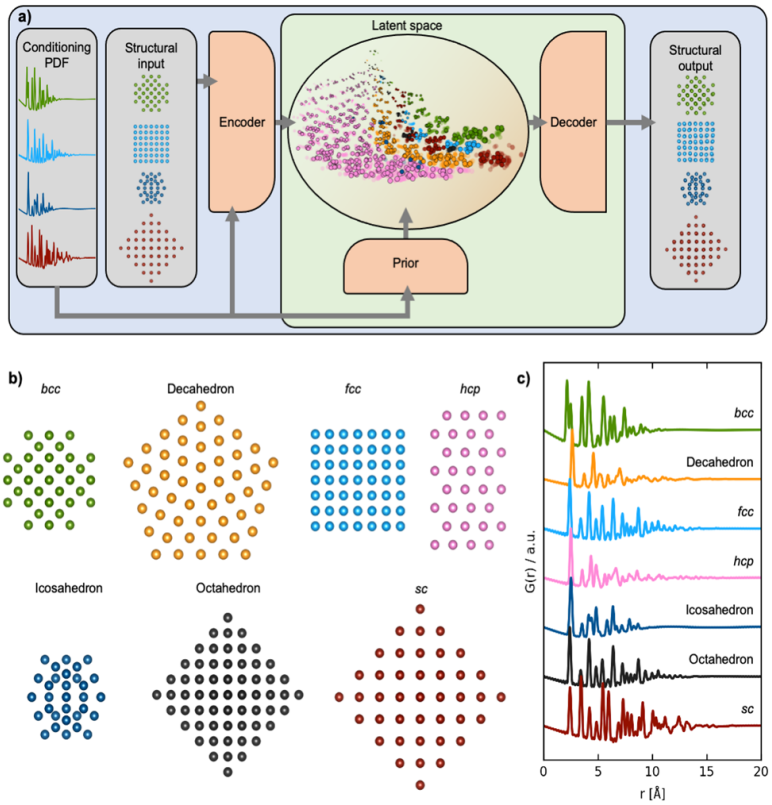

Figure 1: Training DeepStruc to determine the structure of MMNPs from PDFs. a) DeepStruc predicts the xyz-coordinates of the MMNP structure with conditional input provided in the form of a PDF. The encoder uses the structure and its PDF as input while the prior only takes the PDF as input. To obtain the structural output a latent space embedding is given as input to the decoder which produces the corresponding MMNP xyz-coordinates. During training of DeepStruc both the blue and green regions are used, while only the green region is used for structure prediction during the inference process. b) Examples of the seven different structure types which are used as input to DeepStruc together with their c) simulated PDFs used as conditioning in DeepStruc. Each structure type has been included in the training set with varying sizes of 5 to 200 atoms and with varying lattice constants. The 3743 structures were split into training- (60 %), validation- (20 %), and testing sets (20 %).

or expanded compared to the original structure. We do not yet understand why DeepStruc has this aberration, but it is easily solvable by refining an expansion/contraction variable as a post processing step to DeepStruc. After refining the structure to the PDF[34] by fitting a contraction/expansion factor, a scale factor and an isotropic atomic displacement parameter (ADP), as described in section E in the Supplementary Information, the MAE of the atom positions is reduced to 0.093 ± 0.058. The inference is thus robust against moderate changes in lattice parameter between a provided PDF and the structures that DeepStruc were trained on. The reconstructed structures exhibit some artificial positional atomic disorder that broadens the PDF peaks. The fitted ADP values (section E in the Supplementary Information) are thus lower than the ADP values of the conditioning PDFs.

Having established that DeepStruc works for structures highly resembling those in the training set, we now consider more challenging cases and explore the capabilities of DeepStruc on an actual test set which is far from the training distribution. As described above, the largest structures in the training set contained only 200 atoms.

In practice, DeepStruc must be able to yield valid reconstructed structures from experimental data that contain noise and other aberrations. We therefore use DeepStruc to infer structures from previously published experimental PDFs from MMNPs. Fig. 4a shows the latent space with the

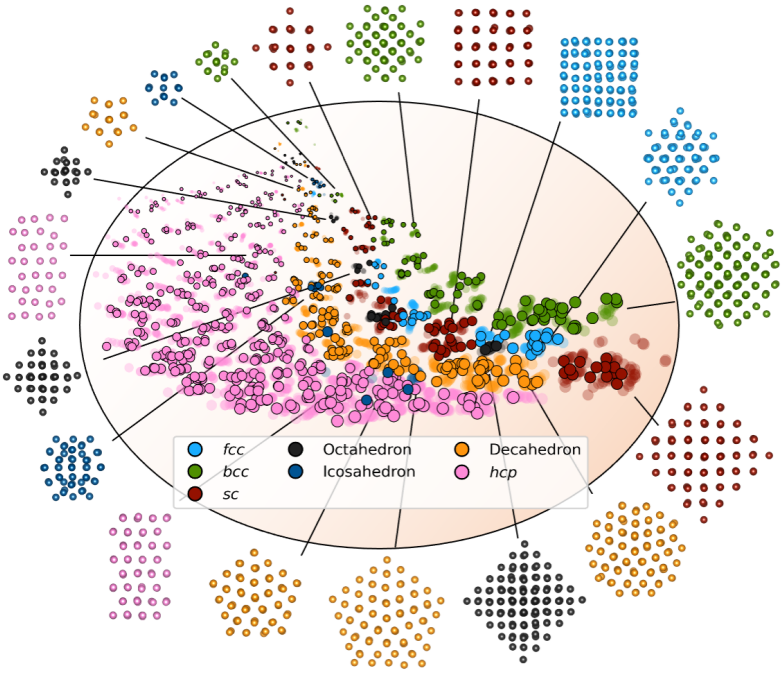

Figure 2: The two-dimensional latent space with structure reconstructions. The points in the latent space correspond to a structure and its simulated PDF. Data points from the test set are shown in solid colour and outlined. The points from the training and validation sets are shown as semi-transparent. The size of the points relates to the size of the embedded MMNP, and the orange background indicates the general size increase throughout the latent space. The colour of each point resemblances its structure type, *fcc* (light blue), octahedral (dark grey), decahedral (orange), *bcc* (green), icosahedral (dark blue), *hcp* (pink), and *sc* (red). Note that the structures shown here are predicted by DeepStruc during inference on PDFs from the test set.

predicted location of structures from three experimental PDFs. Here, the location in the latent space is represented as distributions rather than as discrete points, and multiple structures are sampled from each distribution and compared to the experimental PDF to select the best candidate. The mean of the experimental PDF distributions is represented as a black diamond with three ellipsoids indicating different confidence intervals with σ: 3, 5 and 7, where σ is the standard deviation of the normal distribution.

The first experimental dataset that we evaluate was published by Jensen et al.,[35] who identified a decahedral structure as the core motif of Au_{144}(*p*-MBA)$_{60}$ nanoparticles. DeepStruc locates the Au_{144}(*p*-MBA)$_{60}$ PDF (Fig. 4b) in a decahedral region (orange distributions in Fig. 4a) in the latent space. Given the generative capabilities of DeepStruc, in theory, we can sample an unlimited number of structures for a given PDF. As described in section F of the Supplementary Information, we here sampled up to 1000 structures from the three normal distributions (σ: 3, 5, and 7), and compared their fit to the experimental PDF. Fig. 4b shows the fit of the best structural prediction, which was among the structures sampled from the σ: 3 distributions. DeepStruc predicts a decahedral structure, which agrees well with the literature.[35] Other structures sampled from the three distributions are shown in section G of the Supplementary Information, where we also compare the DeepStruc analysis to baseline methods. We first consider a brute-force structure-mining method inspired by Banerjee et al.,[36], but also compare the DeepStruc results to two simpler ML-algorithms, namely a tree-based ML classifier and a regular CVAE without a graph-based input.

The second dataset that we evaluate, published by Quinson et al.,[37] are from 1.8 nm Pt nanoparticles with the *fcc* structure (described further in Section H in the Supplementary Information). This size corresponds to ca. 203 atoms, i.e. the number of atoms in the particle goes slightly beyond the *fcc* structures in the training set that contain only 165 atoms.[37] The location of the predicted mean is

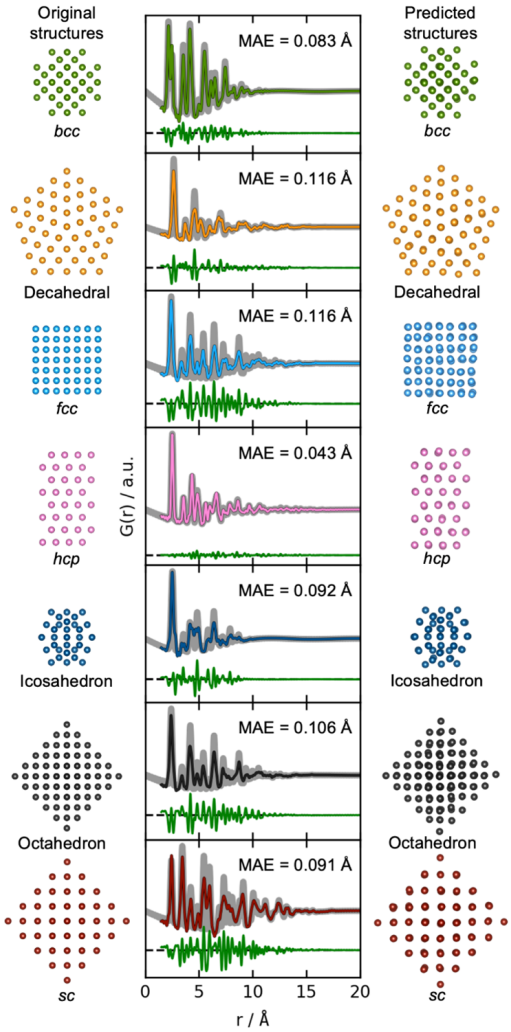

Figure 3: Structure determination from PDFs. Simulated PDFs (grey) from the original structures of the seven different structure types (left) are used during inference for structure prediction (right). The middle column shows the fitted PDFs of the predicted structures to the simulated PDFs of the original structures. Only the scale-factor, contraction/expansion-factor, and ADP are refined, see section E in the Supplementary Information.

again shown as a black diamond in Fig. 4a, enclosed by three blue ellipsoids illustrating different magnitudes of standard deviation. The mean of the predicted structure is placed near the largest *sc* structures. If DeepStruc only favoured symmetry it would be placed directly on the *fcc* structures. Interestingly, DeepStruc does not purely favour size either, as it does not position the PDF near the largest structures which are *hcp* structures of 200 atoms. Instead, we observe that DeepStruc takes both symmetry and size into account by placing the mean predicted structure adjacent to the largest *sc* structures containing 185 atoms. To identify the structure from the experimental PDF, we again sample 1000 structures from the σ: 3, 5 and 7 distributions. When fitting these sampled structures to the dataset, we obtain the best fit from an *fcc* structure of 146 atoms that is visualized in Fig. 4c and which agrees with the baseline models (section G in the Supplementary Information). DeepStruc thus identifies an *fcc* structure even though the size of the MMNP is outside the training set distribution. Additionally, we test the extrapolation capabilities of DeepStruc with both simulated and experimental data. This is described in section I in the Supplementary Information.

While DeepStruc only has been trained on simple MMNPs, we finally evaluate it on a PDF from $Au_{144}(PET)_{60}$ nanoparticles, consisting of an icosahedral core of 54 atoms surrounded by a rhombi-

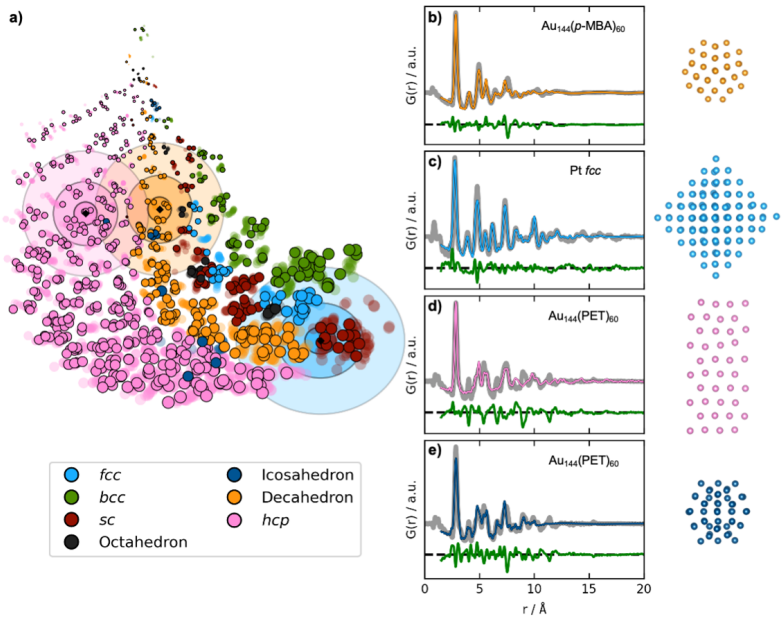

Figure 4: Fitting experimental PDFs with structures obtained by DeepStruc. a) The DeepStruc latent space showing predicted latent space positions for structures from three experimental PDFs. The predicted means are shown as diamond markers, which are enclosed by three rings, indicating the sampling regions for σ: 3, 5, and 7. b) PDF fit of the reconstructed structure from the Au_{144}(*p*-MBA)$_{60}$ PDF[35] c) PDF fit of the reconstructed structure from the 1.8 nm Pt nanoparticle PDF from Quinson et al.[37], d) PDF fit of the reconstructed structure from the Au_{144}(*p*-MBA)$_{60}$ PDF[35] using a *hcp* structure. e) PDF fit of the reconstructed structure from the Au_{144}(*p*-MBA)$_{60}$ PDF[35] using an icosahedral structure. Note that the test set structures shown here are the predicted structures from DeepStruc obtained during inference on experimental PDFs.

cosidodecahedron shell of 60 atoms (Fig. 4d and e).[35,38] We show the predicted mean position of the structure with a black diamond enclosed by pink ellipsoids. DeepStruc positions the PDF in the *hcp* region of the latent space, and when sampling 1000 structures from the distribution with σ: 7, the best fitting structures is an *hcp* structure with 40 atoms for the $Au_{144}(PET)_{60}$ nanoparticle (Fig. 4d). Similar structures are found when sampling from the σ: 3 and σ: 5 distributions. However, the PDF fit reveals that the reconstructed structure does not capture all peaks in the experimental PDF. When considering further the latent space, icosahedral structures are strongly underrepresented in our dataset (section A in the Supplementary Information) which results in an inconsistency when placing icosahedral structures in the latent space. DeepStruc is thus challenged when solving the icosahedral core structure of the nanoparticle. However, we observe that one of the test icosahedral structures is placed near the experimental PDF in latent space within the σ: 5 distribution. Therefore, we again try to sample 1000 structures by moving the mean of the σ: 3 distribution to the nearest cluster of icosahedral structures in the latent space, which are located right outside the σ: 7 distribution. The best fitting structure (Fig. 4e) captures all main peaks of the experimental PDF. Strategies for sampling of underrepresented structures is discussed further in section F in the Supplementary Information.

3.4 Structure determination from PDF: *fcc*, *hcp*, and stacking faulted nanoparticles

To obtain a deeper understanding of the latent space's behaviour, we investigate a dataset only containing *fcc*, *hcp*, and stacking faulted structures. *Fcc* and *hcp* structures are distinguished by the stacking sequence of closed packed layers in their structures: while *fcc* structures can be described by ABCABC stacking, *hcp* structures have ABABAB stacking. Structures with other sequences are stacking faulted structures. We hypothesize that stacking faulted structures can be considered an 'interpolation' in the discrete space between the *fcc* and *hcp* structure type.[30]

Examples of reconstructed *fcc* (blue), *hcp* (pink), and different stacking faulted structures (purple) and their position in the new latent space are illustrated in Fig. S8a. The MMNPs cluster in size, whilst we also observe that *fcc* and *hcp* structures separate in the latent space. It is evident that the stacking faulted structures are located in between the *fcc* and *hcp* structures in the latent space as hypothesized. It is chemically reasonable that they are positioned in this exact order based on their similarity to *fcc* and *hcp*. For example, the structure with ABCABA layers, shown in Fig. S8 with a purple star is structurally close *fcc*. We see that it is also located closer to the *fcc* structures in the latent space. On the other hand, the structure with ABCBCB layers (marked as a purple diamond in Fig. S8) can be considered structurally more closely related to *hcp* than *fcc*. DeepStruc places this structure adjacent to *hcp* structures of the same size in the latent space. DeepStruc can thus insert stacking faulted structures between *fcc* and *hcp* into the latent space in a chemically meaningful way.

4 Discussion

We have shown the potential of using a DGM for structure determination from simulated and experimental PDFs. Our graph-based CVAE algorithm, DeepStruc, provides valuable information through its latent space, as the MMNP structures cluster based on symmetry and size in agreement with their structural chemistry. Using experimental data, the Au_{144}(*p*-MBA)$_{60}$ nanoparticle was determined to be decahedral, Pt nanoparticles were determined to be *fcc* and the Au_{144}(PET)$_{60}$ was determined to have an icosahedral core structure, all in agreement with previous literature. While these systems are relatively simple MMNPs, we recognise that there are more complex materials where the measured PDF would not contain sufficient information to solve the structure. DeepStruc would then be limited to provide a distribution of starting models which can aid in the further structure analysis.

Our approach is only restricted by the distribution of the structural training set. When DeepStruc is trained on *fcc*, *hcp*, and stacking faulted structures, it will locate the stacking faulted structures in between the *fcc* and *hcp* structures. This suggests a strategy for training DeepStruc models on different chemical systems that also 'interpolate' from one to another when this can be identified. DeepStruc does not yet provide a completely general structure solution approach, but gives critical insight into how DGMs can interact with structural and diffraction information to yield candidate structures and ultimately structure solutions.

We suggest to implement DeepStruc as part of PDF-in-the-cloud (PDFitc.org),[39] where the training data can gradually be expanded over time. So far, the structures investigated are fairly ordered and contain some symmetry, but in the future, we plan to expand DeepStruc to chemical systems with more atoms and higher complexity such as metal oxide nanoparticles and alloys. Combining the PDF conditioning with data from complimentary techniques could prove important for structure determination of more complex systems. Such studies would both enable structure determination from a combined modelling perspective, but it would also reveal fundamental aspects of the information content of the different datasets for solving structure problems.

References

[1] David, W. I. F. Shankland, K. Structure determination from powder diffraction data. Acta Crystallogr. A 64, 52-64 (2008).

[2] Cheetham, A. K. Goodwin, A. L. Crystallography with powders. Nat. Mater. 13, 760-762, doi:10.1038/nmat4044 (2014).

[3] Billinge, S. J. L. Kanatzidis, M. G. Beyond crystallography: the study of disorder, nanocrystallinity and crystallographically challenged materials with pair distribution functions. Chem. Commun., 749-760, doi:Doi 10.1039/B309577k (2004).

[4] Young, C. A. Goodwin, A. L. Applications of pair distribution function methods to contemporary problems in materials chemistry. J. Mater. Chem. 21, 6464-6476, doi:10.1039/C0JM04415F (2011).

[5] Christiansen, T. L., Cooper, S. R. Jensen, K. M. Ø. There's no place like real-space: elucidating size-dependent atomic structure of nanomaterials using pair distribution function analysis. Nanoscale Adv. 2, 2234-2254, doi:10.1039/D0NA00120A (2020).

[6] Zhu, H. et al. Bridging Structural Inhomogeneity to Functionality: Pair Distribution Function Methods for Functional Materials Development. Adv. Sci. 8, 2003534, doi:https://doi.org/10.1002/advs.202003534 (2021).

[7] Billinge, S. J. L. Levin, I. The problem with determining atomic structure at the nanoscale. Science 316, 561-565, doi:Doi 10.1126/Science.1135080 (2007).

[8] Yang, L., Juhas, P., Terban, M. W., Tucker, M. G. Billinge, S. J. L. Structure-mining: screening structure models by automated fitting to the atomic pair distribution function over large numbers of models. Acta Crystallogr. A 76, 395-409, doi:doi:10.1107/S2053273320002028 (2020).

[9] Banerjee, S. et al. Cluster-mining: an approach for determining core structures of metallic nanoparticles from atomic pair distribution function data. Acta Crystallogr. A 76, 24-31, doi:doi:10.1107/S2053273319013214 (2020).

[10] Christiansen, T. L. et al. Structure analysis of supported disordered molybdenum oxides using pair distribution function analysis and automated cluster modelling. J. Appl. Crystallogr. 53, 148-158, doi:doi:10.1107/S1600576719016832 (2020).

[11] Juhás, P., Cherba, D. M., Duxbury, P. M., Punch, W. F. Billinge, S. J. L. Ab initio determination of solid-state nanostructure. Nature 440, 655-658, doi:10.1038/nature04556 (2006).

[12] Juhás, P., Granlund, L., Duxbury, P. M., Punch, W. F. Billinge, S. J. The Liga algorithm for ab initio determination of nanostructure. Acta Crystallogr. A 64, 631-640, doi:10.1107/s0108767308027591 (2008).

[13] Juhas, P., Granlund, L., Gujarathi, S. R., Duxbury, P. M. Billinge, S. J. Crystal structure solution from experimentally determined atomic pair distribution functions. J. Appl. Crystallogr. 43, 623-629 (2010).

[14] Cliffe, M. J., Dove, M. T., Drabold, D. Goodwin, A. L. Structure determination of disordered materials from diffraction data. Phys. Rev. Lett. 104, 125501 (2010).

[15] Cliffe, M. J. Goodwin, A. L. Nanostructure determination from the pair distribution function: a parametric study of the INVERT approach. J. Phys.: Condens. Matter 25, 454218 (2013).

[16] Choudhary, K. et al. Recent Advances and Applications of Deep Learning Methods in Materials Science. arXiv, 2110.14820 (2021).

[17] Razavi, A., Van den Oord, A. Vinyals, O. Generating diverse high-fidelity images with vq-vae-2. Adv. Neural Inf. Process. Syst. 32 (2019).

[18] Karras, T. et al. Analyzing and improving the image quality of stylegan. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 8110-8119 (2020).

[19] Hjorth Larsen, A. et al. The atomic simulation environment—a Python library for working with atoms. J. Phys.: Condens. Matter 29, 273002, doi:10.1088/1361-648x/aa680e (2017).

[20] Anker, A. S. et al. Characterising the Atomic Structure of Mono-Metallic Nanoparticles from X-Ray Scattering Data Using Conditional Generative Models. Proceedings of the 16th International Workshop on Mining and Learning with Graphs (MLG), doi:10.26434/chemrxiv.12662222.v1 (2020).

[21] Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A. Vandergheynst, P. Geometric deep learning: going beyond euclidean data. IEEE Signal Process. Mag. 34, 18-42 (2017).

[22] Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M. Monfardini, G. The graph neural network model. IEEE Trans. Neural Netw. 20, 61-80 (2008).

[23] Kingma, D. P. Welling, M. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013).

[24] Sohn, K., Lee, H. Yan, X. Learning structured output representation using deep conditional generative models. Adv. Neural Inf. Process. Syst. 28, 3483-3491 (2015).

[25] Lim, J., Ryu, S., Kim, J. W. Kim, W. Y. Molecular generative model based on conditional variational autoencoder for de novo molecular design. J. Cheminformatics 10, 1-9 (2018).

[26] Duxbury, P. M., Granlund, L., Gujarathi, S., Juhas, P. Billinge, S. J. The unassigned distance geometry problem. Discret. Appl. Math. 204, 117-132 (2016).

[27] Shao, H. et al. ControlVAE: Tuning, Analytical Properties, and Performance Analysis. IEEE Trans. Pattern Anal. Mach. Intell. (2021).

[28] Rydhmer, K. Selvan, R. Dynamic β-VAEs for quantifying biodiversity by clustering optically recorded insect signals. arXiv preprint arXiv:2102.05526 (2021).

[29] Kipf, T. N. Welling, M. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016).

[30] Samarakoon, A. M. et al. Machine-learning-assisted insight into spin ice Dy2Ti2O7. Nat. Commun. 11, 892, doi:10.1038/s41467-020-14660-y (2020).

[31] Gómez-Bombarelli, R. et al. Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. ACS Cent. Sci. 4, 268-276, doi:10.1021/acscentsci.7b00572 (2018).

[32] Marks, L. D. Surface structure and energetics of multiply twinned particles. Philos. Mag. A 49, 81-93, doi:10.1080/01418618408233431 (1984).

[33] Banerjee, S. et al. Improved Models for Metallic Nanoparticle Cores from Atomic Pair Distribution Function (PDF) Analysis. J. Phys. Chem. C 122, 29498-29506, doi:10.1021/acs.jpcc.8b05897 (2018).

[34] Juhas, P., Farrow, C. L., Yang, X., Knox, K. R. Billinge, S. J. L. Complex modeling: a strategy and software program for combining multiple information sources to solve ill posed structure and nanostructure inverse problems. Acta Crystallogr. A 71, 562-568, doi:doi:10.1107/S2053273315014473 (2015).

[35] Jensen, K. M. Ø. et al. Polymorphism in magic-sized Au144(SR)60 clusters. Nat. Commun. 7, doi:10.1038/ncomms11859 (2016).

[36] Banerjee, S. et al. Cluster-mining: an approach for determining core structures of metallic nanoparticles from atomic pair distribution function data. Acta Crystallographica Section A: Foundations and Advances 76, 24-31 (2020).

[37] Quinson, J. et al. Spatially Localized Synthesis and Structural Characterization of Platinum Nanocrystals Obtained Using UV Light. ACS Omega 3, 10351-10356, doi:10.1021/acsomega.8b01613 (2018).

[38] Yan, N. et al. Unraveling the long-pursued Au144 structure by x-ray crystallography. Sci. Adv. 4, eaat7259, doi:doi:10.1126/sciadv.aat7259 (2018).

[39] Yang, L. et al. A cloud platform for atomic pair distribution function analysis: PDFitc. Acta Crystallographica Section A 77, 2-6, doi:doi:10.1107/S2053273320013066 (2021).