

ARTICLE OPEN



Tracking perovskite crystallization via deep learning-based feature detection on 2D X-ray scattering data

Vladimir Starostin¹, Valentin Munteanu¹, Alessandro Greco¹, Ekaterina Kneschaurek¹, Alina Pleli¹, Florian Bertram², Alexander Gerlach¹, Alexander Hinderhofer¹ and Frank Schreiber¹

Understanding the processes of perovskite crystallization is essential for improving the properties of organic solar cells. In situ real-time grazing-incidence X-ray diffraction (GIXD) is a key technique for this task, but it produces large amounts of data, frequently exceeding the capabilities of traditional data processing methods. We propose an automated pipeline for the analysis of GIXD images, based on the Faster Region-based Convolutional Network architecture for object detection, modified to conform to the specifics of the scattering data. The model exhibits high accuracy in detecting diffraction features on noisy patterns with various experimental artifacts. We demonstrate our method on real-time tracking of organic-inorganic perovskite structure crystallization and test it on two applications: 1. the automated phase identification and unit-cell determination of two coexisting phases of Ruddlesden–Popper 2D perovskites, and 2. the fast tracking of MAPbI₃ perovskite formation. By design, our approach is equally suitable for other crystalline thin-film materials.

npj Computational Materials (2022)8:101; <https://doi.org/10.1038/s41524-022-00778-8>

INTRODUCTION

Perovskite materials have been heralded as the future of solar cell technology, promising low cost and high efficiency^{1,2}. In the last decade, there has been an enormous interest in enabling the commercial use of perovskite solar cells by optimizing their structural and optoelectronic properties. To this end, a detailed understanding of their crystallization pathways is essential. Grazing-incidence X-ray diffraction (GIXD, which we here consider equivalent to grazing-incidence wide-angle scattering, GIWAXS) is a key technique for this task, enabling real-time, non-destructive probing of perovskite crystal structures during their synthesis^{3–7}. Currently, thanks to advances in detector technology, the amount of collected raw data from in situ GIXD experiments reaches millions of images per measurement day, far exceeding the capabilities of traditional data processing methods. As a result, only a tiny portion of the collected data is analyzed, so that important processes remain hidden and undiscovered. Therefore, the full potential of the technique is not exploited, and novel, fast technologies for data analysis are required⁸.

One of the solutions is to employ machine learning which has already proven advantageous for the analysis of various kinds of scattering data^{9–15}. In particular, there is an increasing interest in data-driven approaches for material science, such as automated phase identification and unit-cell determination from one-dimensional X-ray powder diffraction (XRD) measurements^{13–17}. However, there are no corresponding methods for GIXD analysis yet. The existing machine learning solutions for two-dimensional diffraction data are mainly focused on the preliminary classification of the images^{18–20} and refining positions of stand-alone peaks for semi-automated data processing^{21,22}. The more profound tasks, such as phase identification, determination of preferred orientations and unit cell parameters, require careful extraction of all diffraction feature positions. Once the peak positions are established, one can perform a comprehensive analysis of the diffraction patterns autonomously via existing

algorithms for peak indexing, matching, unit-cell determination. Therefore, the peak detection procedure is the key bottleneck on the way to automated GIXD analysis. Furthermore, peak detection is also suitable for fast preliminary filtering of the data during the experiment.

In this work, we implemented a modern two-stage deep learning object detection approach for fast and accurate Debye-Scherrer ring and segment location in GIXD images. As a model, we chose the Faster Region-based Convolutional Network (R-CNN) deep learning architecture^{23,24}, for which we introduced several essential modifications to conform to the specifics of GIXD diffraction data and increase the performance. We trained the model on simulated images via a simulation procedure aimed at reproducing variate scattering backgrounds and experimental artifacts. We note that the translation-invariant Faster R-CNN algorithm detects each peak independently; thus, our approach is material-agnostic by design, unlike in the case of a model trained to extract material-related information from the measured data at a single step.

As a demonstration of the possible applications of our method, we performed an automated analysis on two in situ GIXD datasets of organic-inorganic perovskite structures. However, we emphasize that it is, in fact, generally applicable to essentially any crystalline thin-film materials. First, we track the formation of different phases in 2D halide perovskites. Our model detects diffraction reflections, which are then used to identify the perovskite phases and crystal orientations and track the evolution of the lattice parameters over time, which is crucial for the photovoltaic performance. Second, we demonstrate a fast analysis of over a thousand diffraction images from in situ measurements of the conversion of methylammonium lead iodide (MAPbI₃) from an intermediate phase during spin-coating and annealing. The algorithm detects over 20,000 peaks on 1015 images and can pinpoint the start of perovskite formation during the experiment.

¹Institute of Applied Physics, University of Tübingen, 72076 Tübingen, Germany. ²Deutsches Elektronen-Synchrotron DESY, Notkestr. 85, 22607 Hamburg, Germany. ✉email: vladimir.starostin@uni-tuebingen.de; alexander.hinderhofer@uni-tuebingen.de; frank.schreiber@uni-tuebingen.de

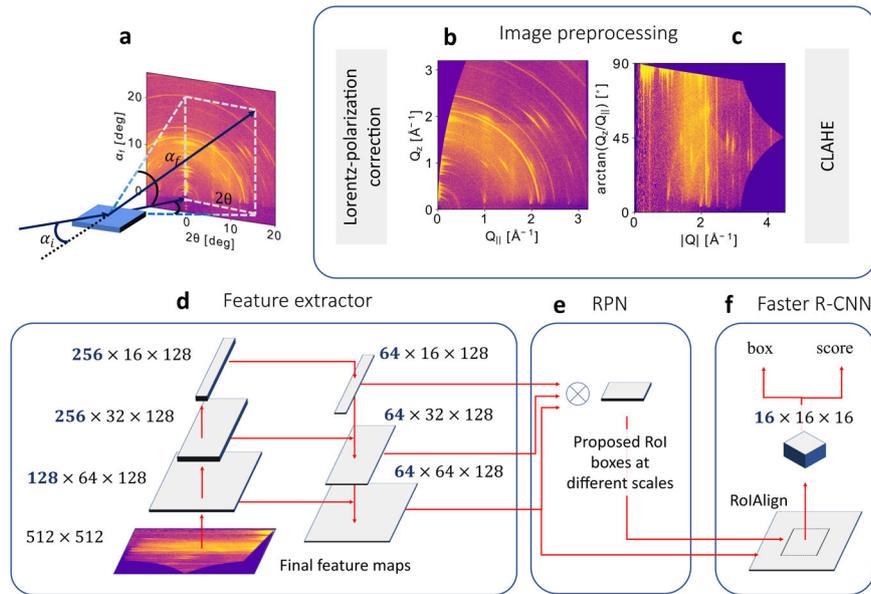


Fig. 1 Image preprocessing pipeline and the model architecture. **a** The geometry of the measurements. After Lorentz-polarization correction, measured diffraction patterns are sequentially converted from detector coordinates (**a**) to reciprocal space (**b**) and then to polar coordinates (**c**). For the detection, the contrast is enhanced by CLAHE (all shown images are already contrast-enhanced for visualization). **d** Feature extractor with asymmetric feature maps; feature shapes correspond to an input image size of 512×512 pixels. **e** The Region Proposal Network kernels convolve with feature maps to extract RoI at different scales. **f** At the second detection stage, a RoIAlign layer extracts features at corresponding positions from the largest feature map, from which the box coordinates and the score of confidence are predicted for each box by a fully-connected network.

RESULTS

Data-driven model modifications

We implemented a deep learning object detection model based on a two-stage Faster R-CNN framework^{23,24} to locate diffraction rings and segments in GIXD images. Figure 1 illustrates the data pipeline with the model architecture (see Methods IV B). State-of-the-art object detection algorithms are mainly focused on improving performance on objects in RGB photographs and the corresponding benchmark datasets (PASCAL VOC, MS COCO, Open Images, etc.)^{23,25–28}. X-ray scattering patterns exhibit particular features that require careful adjustments of the existing general methods. To this end, we introduced an array of modifications to the original architecture to conform to the specifics of the scattering data and accelerate the calculations. In the following, we briefly discuss the key features of the grazing-incidence diffraction images that inform the design of the detection algorithm.

Polar symmetry. X-ray scattering from crystalline grains results in diffraction patterns showing either full rings or ring segments in reciprocal space (depending on their orientation distribution) (see Fig. 1b). However, most detection algorithms are designed to detect an object by providing coordinates of a rectangular bounding box around the object. The rectangular shape is most suited for feature extractors that are based on convolution operations with rectangular kernels, while detecting objects with circular shapes would lead to nontrivial complications. To overcome these issues and work with rectangular object shapes, we converted the GIXD images to the polar coordinates $|Q| = (Q_x^2 + Q_y^2)^{\frac{1}{2}}$ and $\phi = \arctan(Q_x/Q_y)$, where Q is the scattering vector and $Q_{||} = (Q_x^2 + Q_y^2)^{\frac{1}{2}}$. We note that some effects such as refraction may distort the polar symmetry in grazing incidence geometry, but they can be largely neglected here.

Simple features with complex experimental artifacts. In general, Bragg peak profiles can be well approximated by a Voigt function or similar profiles (Gaussian, Pearson VII, pseudo-Voigt, etc.^{29–31}). At the same time, Bragg peaks are typically influenced by various experimental artifacts: incoherent scattering, scattering from amorphous substrates and gases, counting statistics, the direct beam, detector gaps, etc. In this light, we conclude that a deep learning-based solution is required to filter out artifacts and provide stable detection accuracy. However, in contrast to detecting more complex objects in RGB photographs (e.g., humans, animals, cars), a less deep representation might be sufficient for detecting Bragg reflections. Moreover, we can omit a classifier from the detection algorithm, since there is no task to classify peaks by their shape. These simplifications significantly accelerate the model, which is desirable for on-the-fly analysis of measurements with high acquisition rates (up to tens of kHz on modern detectors³²).

"Fractal" properties of the detected objects. In terms of detection, diffraction rings exhibit properties unusual for objects in RGB photographs. For example, it is possible that a segment of a ring is detected with a higher confidence score than the whole ring. At the same time, two separate but adjacent segments at the same $|Q|$ may be considered a whole ring on a compressed feature map. This behavior causes corresponding detection errors and requires certain adjustments to the training process that are discussed in Methods IV B.

Asymmetric object shapes at different scales. Debye-Scherrer rings are typically well-localized in the radial dimension (horizontal in polar coordinates) (see Fig. 1c). However, their sizes along the angular (vertical) axis may vary substantially depending on the distribution of orientations of the crystalline grains. Modern deep convolutional feature extractors are designed to decrease the spatial resolution in exchange for richer feature maps that may distinguish between different complex shapes. This strategy

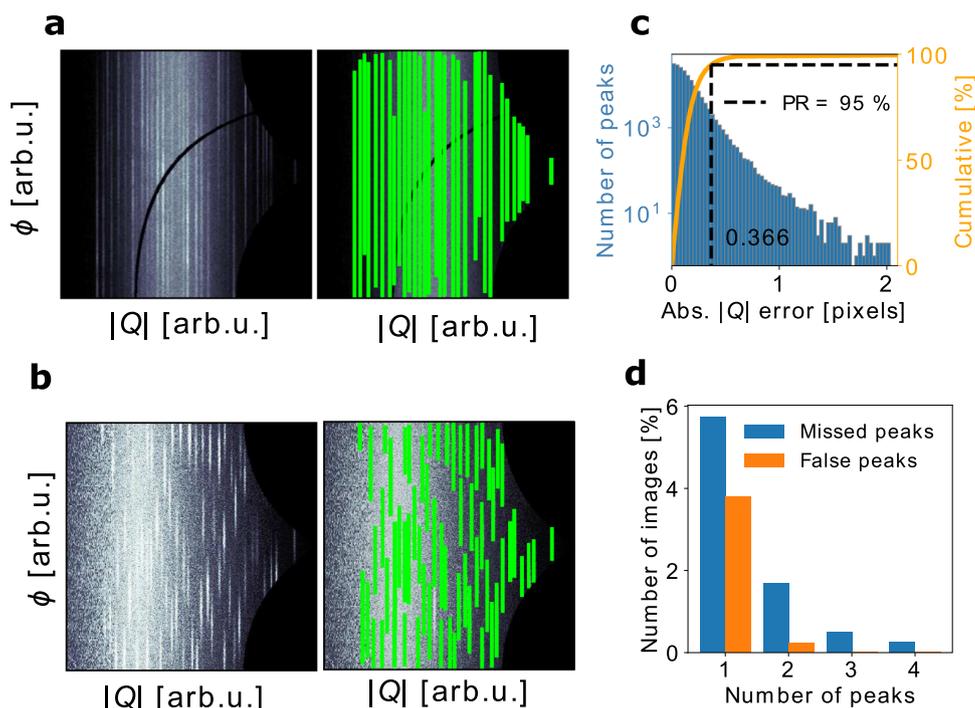


Fig. 2 Detection results on the simulated test set of 10,000 images. **a** and **b** Show two simulated patterns (left) and the corresponding detection predictions (right) in the polar coordinates $|Q| = (Q_z^2 + Q_{||}^2)^{\frac{1}{2}}$ and $\phi = \arctan(Q_z/Q_{||})$. **c** The absolute $|Q_{\text{detected}}$ error distribution with cumulative curve; the error is below 0.366 pixels for 95% of the peaks. **d** Distribution of missed and falsely detected peaks per image.

would be ill-suited for our task as we would prefer to keep a sufficient resolution along the horizontal $|Q|$ axis while compressing the features along the vertical axis. Therefore, we use asymmetric convolutional operations to address the asymmetric shapes of the detected object and to preserve a high resolution along the $|Q|$ axis. We also use several feature maps of different shapes to identify peaks at various scales (see Methods IV B).

High dynamic range. Modern X-ray area detectors can register up to billions of counts per pixel³² creating exceptionally large dynamic ranges. In practice, such bright spots may correspond to strong reflections from rare large crystalline grains or from the direct beam profile. At the same time, other features may have orders of magnitude less counts, especially at higher Q values. This range in contrast makes it challenging for the neural network to equally detect peaks of different brightness. Therefore, we chose to contrast-enhance the experimental images as a preprocessing step (see Methods IV A).

Performance on the simulated data

The model was trained (see Methods IV D) on a training set of 48,000 simulated images which emulate GIXD images in polar space, encompassing a highly diverse configuration of Bragg reflections, scattering backgrounds and experimental artifacts (see Methods IV C and Supplementary Fig. 1). The performance of the trained model was first evaluated on a test set of 10,000 simulated images. Figure 2a, b show two examples of the simulated patterns. The green lines correspond to the peaks detected by the model. There were 175,344 peaks simulated for 10,000 images in total, of which 99.22% (173,990) were detected correctly, 0.78% (1354) were missed and 0.25% (439) of the detected peaks were false positives. As a performance metric (Fig. 2c, d) we chose to simply use the absolute peak center distance $\Delta|Q| = ||Q_{\text{detected}}| - |Q_{\text{truth}}||$ over the more commonly used metric Intersection over Union (IoU)^{23,25–27}. This was done because we deemed the accuracy of the peak center to be more relevant to this task than the peak

overlap. Figure 2c shows the distribution of $\Delta|Q|$ for 173,990 peaks along with the cumulative curve; 95% of the peaks were detected with a $\Delta|Q|$ below 0.366 pixels. A stochastic nature of the simulation process (see Methods IV C) unavoidably creates a small fraction of peaks which are unrecognizable due to noise, background, low intensity, etc. Figure 2d illustrates the distributions of falsely detected and missed peaks per image; only 4% of images contained at least one falsely detected peak and 8.5% of images had at least one missed peak. The number of false and missed peaks per image determines the correctness of structure determination. In our case, the vanishingly low numbers of false positive detections per image (0.044 peaks on average) and missed peaks per image (0.14 peaks on average) minimize the corresponding errors in the further analysis of GIXD images.

We note that our neural network is lightweight (5.9 M parameters) and fast (122 images per second), which is essential for real-time GIXD data analysis. For comparison, the analogous unmodified Faster R-CNN model²⁴ is substantially larger (41 M parameters) and slower (26 images per second), yet it misses 7% of the peaks on the test set. This result illustrates that our model is highly optimized for the task and outperforms much larger standard architectures. Furthermore, to justify the choice of the object detection algorithm and the importance of the second detection stage, we provide performance comparisons with two other types of object detection models: the one-stage detector and the segmentation model U-Net³³. The results are summarized in Supplementary Fig. 2 (see Methods IV F for details). Both models exhibit lower detection accuracy than our two-stage detector. Compared to the one-stage detector, the second detection stage is particularly relevant for filtering out false positive predictions (the share of false positives for the one-stage detector is 5.6%). The segmentation model is indeed simpler in terms of implementation; however, in addition to a higher number of false positives (9.6%), it suffers from other systematic problems: the overlapping peaks cannot be separated, and single peaks can be detected as multiple peaks (see Supplementary Fig. 3); the

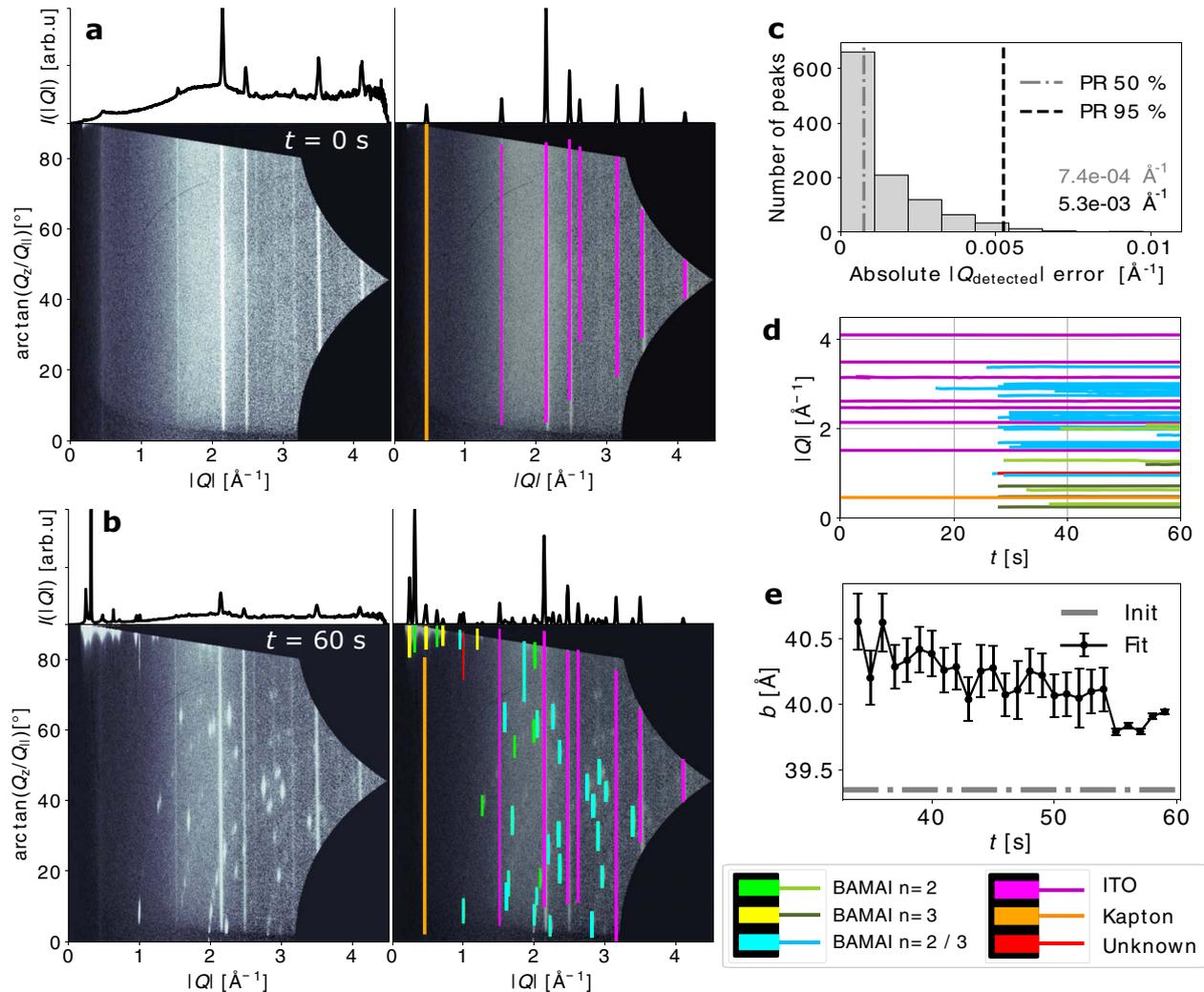


Fig. 3 Detection results of in situ measurements of 2D halide perovskites formation. Diffraction patterns in (a) and (b) are annotated with Bragg reflections detected by the machine learning algorithm at $t = 0$ s and $t = 60$ s, respectively. Radial profiles above the images show the initial radial profiles calculated from the diffraction patterns (left images) and the profiles calculated based on the detected peaks (right images). **c** The distribution of the absolute error of radial peak positions location by the model with 50% and 95% percentiles. **d** The radial positions of the detected peaks over time. **e** The evolution of a lattice parameter b over time for $n = 2$ phase of $(\text{BA})_2(\text{MA})_{n-1}\text{Pb}_{n+1}$ (denoted as BAMA1 on the legend) perovskite calculated via the detected peaks positions. The error bars indicate the standard errors (s.e.m.) of the fit calculated using Hessian matrices.

accuracy of peak $|Q|$ determination is limited by pixel size; the process of extracting peak positions from segmentation maps increases the inference time to 30 images per second.

In the light of these results, it is important to note that, although GIXD data is much harder for humans to interpret than the RGB photographs from typical benchmark object detection datasets, the presented model reached near-optimal performance on the test set. The average precision exceeded 99%, a value which has not yet been reached on datasets with RGB photographs with the current state-of-the-art results being around 60%³⁴. These results provide evidence that the developed model architecture is well suited for the current task. This success inspired us to apply our strategy to specific scientific problems and automatize the experimental data processing, as explained below.

Use case 1: 2D perovskite lattice parameter refinement

In the following, we apply the developed model to analyze data from the in situ GIXD measurements of the formation of a Ruddlesden–Popper 2D perovskite (butylammonium

methylammonium lead iodide $(\text{BA})_2(\text{MA})_{n-1}\text{Pb}_{n+1}$ ^{35–37}) during annealing (Supplementary Fig. 6).

The experiment was performed at the synchrotron radiation source PETRA III, at beamline P08³⁸. The X-ray beam energy was $E = 18$ keV and the angle of incidence was $\alpha_i = 0.5^\circ$, which is above the critical angle for the substrate $\alpha_c = 0.16^\circ$. The investigated thin-film sample was obtained by spin-coating a solution of butylammonium iodide (BAI), methylammonium iodide (MAI) and lead iodide (PbI_2) dissolved in a dimethylformamide and dimethyl sulfoxide mixture (DMF:DMSO = 1:4), on a glass substrate coated with indium tin oxide (ITO) (Supplementary Fig. 4). An IR lamp mounted at the top of the spin coating chamber was used to anneal the sample (see Supplementary Fig. 5). Sixty diffraction patterns with 1 s exposure time each were analyzed, corresponding to 60 s of annealing.

Figure 3 a, b demonstrate the detection results of the neural network on two diffraction patterns for $t = 0$ s and $t = 60$ s. The lines on the right-hand images correspond to the locations of the detected peaks. Our algorithm detected most of the visible reflections - 1156 diffraction rings and segments from 60 time

frames remained after the filtering stage (see Methods IV E). The missed reflections typically have low intensities. Some detection inaccuracies in determining peak angular sizes (in particular, lower edges of detected boxes do not cover the whole peaks) may be related to angular profile dissimilarities between the experimental and simulated peaks.

To measure the accuracy of the obtained peak positions, we extracted one-dimensional radial profiles of each peak based on their predicted locations and fitted them with a Gaussian function with linear background (Eq. 1) via the Levenberg–Marquardt algorithm³⁹:

$$I(|Q|) = I_0 \exp\left(-\frac{(|Q| - Q_{\text{fit}})^2}{2w^2}\right) + B|Q| + C \quad (1)$$

Figure 3c shows the distribution of the absolute error of the detected positions $|Q_{\text{detected}}|$ with respect to Q_{fit} obtained by the fit; 95% of the errors are below $5.3 \times 10^{-3} \text{ \AA}^{-1}$. The colors of the peaks in Fig. 3a, b denote the result of the indexing algorithm used for the identification of the Bragg reflections (see the legend on Fig. 3). We note that our detection algorithm can be coupled with any standard indexing algorithm for diffraction patterns, most (if not all) of which require careful peak position extraction. In this case, it is sufficient to use a simple algorithm for phase identification based on the share of simulated intensities (structure factors) that appear within any of the detected boxes (see Methods IV G for more details). We expect some of the possible phases of $(\text{BA})_2(\text{MA})_{n-1}\text{Pb}_n\text{I}_{3n+1}$ to emerge during annealing, where $n \in [1, 7]$ defines the perovskite layer thickness³⁷. Supplementary Fig. 7 shows that two phases $n=2$ and $n=3$ with (010) orientation are identified by our algorithm. The algorithm also identified seven peaks from the ITO substrate and one peak from the kapton window.

Once the phases are identified, each peak is automatically indexed (one peak remains unidentified and likely belongs to a side product of the annealing). Figure 3d shows radial peak positions over time, from which we can obtain the moment when both perovskite phases emerge ($t \approx 28$ s). One peak was assigned as belonging to both $n=2$ and $n=3$ phases and emerging earlier at $t = 17$ s, however, it is in fact a weak reflection from the ITO substrate at $Q = 2.91 \text{ \AA}^{-1}$.

Supplementary Fig. 8 demonstrates a systematic change of the relative radial position of a perovskite peak over time that might indicate changes in the lattice structure. To study these changes, we performed the unit cell refinement procedure (see Methods IV H) for both phases for each time frame. Figure 3e demonstrates

the behavior of the lattice parameter b of $n=2$ phase that slowly decreases over time from 40.5 \AA , which is about 1 \AA above the previously reported value³⁵ that was used as an initial value for the fitting. The lattice length b of the $n=3$ phase is constant ($b \approx 52.5 \text{ \AA}$) and slightly above the initial value 51.96 \AA . This discrepancy is most likely related to the increased temperature during annealing resulting in asymmetric thermal expansion in the direction corresponding to the spacer molecule. The other parameters (see Supplementary Fig. 9) remain unchanged and are in agreement with the initial values.

Note that while the demonstrated analysis can be easily reviewed and adjusted by an expert at each step if necessary, in general there is no need for a manual input. This way, our detection model allows to automate the essential steps of GIXD data analysis from phase identification and peak indexing to refining the unit cell parameters. Performing the same analysis with conventional tools would require a manual selection and fitting of the peaks for each time frame, which is often so time-consuming that in practice just a small fraction of the obtained reflections is fitted for further analysis. In contrast, our approach potentially allows a more comprehensive analysis already during the experiment.

Use case 2: In situ tracking of MAPbI₃ perovskite formation

As a second use case for our technique, we analyze 1015 diffraction images from an in situ spin-coating and annealing experiment of the formation of 3D MAPbI₃ perovskite. The experimental setup was identical to the one described for use case 1. The thin-film sample was obtained by spin-coating a solution of MAI and PbI₂ dissolved in a mixture of dimethylformamide and dimethyl sulfoxide (DMF:DMSO = 4:1), on a glass substrate coated with ITO.

Figure 4 illustrates the diffraction pattern at $t = 700$ s during annealing (left image) and the corresponding detection results (right image). The model detected all the visible peaks in the pattern, which correspond to the reflections from MAPbI₃, the ITO substrate, PbI₂, the intermediate structure $(\text{MA})_2(\text{DMSO})_2\text{Pb}_3\text{I}_8$, and the kapton window. The indexing step was performed in the same way as in the previous case II C. Figure 4b shows the $|Q|$ positions of the detected peaks over time. In total, there were 20,756 peaks left after the filtering stage (see Methods IV E). This fast preliminary analysis revealed conversion from the precursor phase $(\text{MA})_2(\text{DMSO})_2\text{Pb}_3\text{I}_8$ to MAPbI₃ perovskite that happens from $t \approx 220$ s to $t \approx 300$ s.

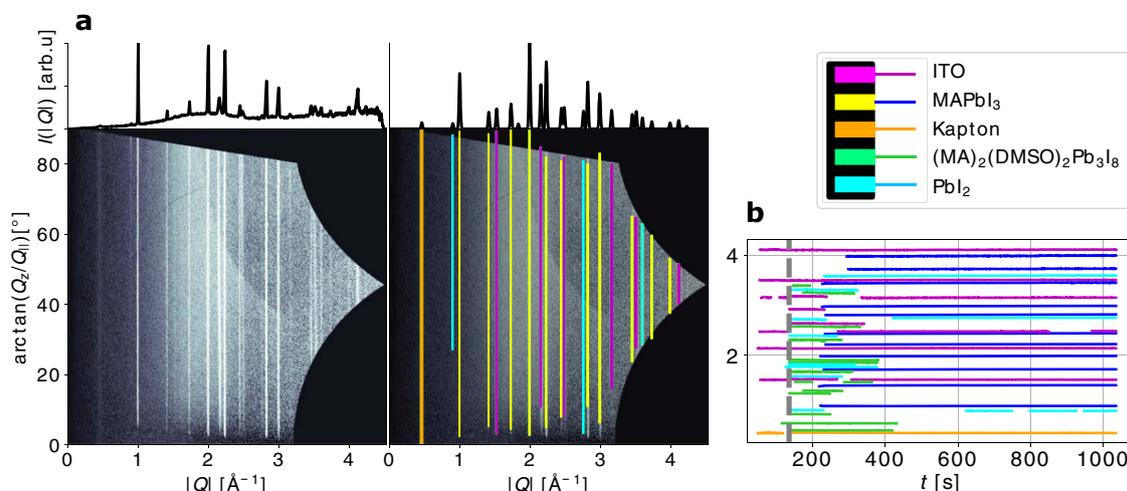


Fig. 4 Detection results of in situ measurements of MAPbI₃ formation during annealing. **a** The diffraction pattern with the detected peaks at $t = 700$ s. **b** The radial positions of the detected peaks over time. The dashed gray line separates the spin-coating and the annealing phases.

DISCUSSION

We demonstrated an automated deep learning approach for the analysis of GIXD images. We showed how to optimize modern object detection algorithms to address various properties of the experimental data, such as the specific geometry, strong background, asymmetry of the diffraction features, etc. These improvements enabled high performance on diverse experimental data with accurate detection of most of the reflections. The small absolute error of the radial peak position determination and high detection accuracy made it possible to perform an in-depth analysis of in situ GIXD data. By processing the extracted feature positions, we identified the formation of two coexisting phases of 2D organic-inorganic perovskite structure and traced the evolution of the lattice parameters in time for these phases. The refinement procedure revealed a slight decrease in the unit cell size for one of the identified phases. We note that such subtle processes may often be overlooked during a manual analysis of the data.

Currently, there is an increasing interest in automated analysis of diffraction data^{13–15,17,20,40,41}. An often proposed solution is a single analysis step where the measured raw data is provided to the machine learning algorithm which is supposed to learn which parts of the data carry information, how to perform the desired analysis and obtain material-related results. This approach needs to be tailored as a whole to variations in materials or experimental setups and does not allow to understand or control the process of such an analysis. In contrast, the more flexible analysis approach presented here consists of modular and transparent parts that are simple to control and adjust for a concrete task.

In general, our solution allows to substantially accelerate the analysis process of GIXD images, potentially boosting the speed of scientific discoveries in material science and organic photovoltaics. The other possible applications of the method include the real-time adjustment of the experimental conditions based on the obtained GIXD data and the high-throughput screening of possible perovskite compositions for organic solar cells.

METHODS

Image preprocessing

Figure 1a–c demonstrate the individual steps of the preprocessing pipeline applied to all the experimental data used in this work. The measured pattern is first corrected by the Lorentz-polarization factor⁴², converted to reciprocal space ($Q_{||}$, Q_{\perp}) using the knowledge about the experimental geometry and mapped to polar coordinates ($|Q|$, $\arctan(Q_{\perp}/Q_{||})$) with an image resolution 512×1024 pixels. The image resolution can be extended for larger area detectors without any adjustments to the method. Finally, the images are contrast-enhanced via the contrast-limited adaptive histogram equalization algorithm (CLAHE)⁴³. The contrast-enhanced images are only used for the detection stage, but not for any further analysis.

Model architecture

For the detection, we chose a two-stage Faster R-CNN object detection framework^{23,24,26} and modified it to meet the specifics of the data discussed above. The general pipeline of the detection process is illustrated by the scheme in Fig. 1. First, a convolutional neural network (feature extractor) is applied to an image to convert it into several feature maps with reduced spatial resolution that, in general, carry information about the appearance of different objects on an image. During the first detection stage, a Region Proposal Network (RPN) slides over the feature maps via the convolution operation. For each pixel on each feature map, it determines whether there is any object at any of the predefined scales (anchors²³) by providing objectness score and the box coordinates relative to the anchors. The resulting regions of interest (RoI) with positive objectness score are individually extracted from the feature maps, reshaped and provided to a fully-connected network²⁵ that classifies an object and refines its box coordinates. For more details on the basic

implementation we refer to^{23,24}, and below we focus on the key differences determined by the specifics of the data.

Feature extractor. We design a feature extractor with asymmetric convolutional layers to preserve resolution along the radial axis $|Q|$. For this purpose, we modify a small ResNet-18 model pretrained on ImageNet dataset⁴⁴ by using asymmetric stride = (2, 1) for the three last residual blocks of the model. The use of the pretrained model allows speeding up the training (see Supplementary Fig. 10). Thus, for an input image of 512×512 pixels, the output feature maps from these three blocks would have shapes of 64×128 , 32×128 and 16×128 , respectively. In this way, an image size is mostly reduced in the vertical direction to allow the detection of long narrow rings. At the same time, vertical size reduction from 512 to 16 pixels makes it impossible to resolve segments with small angular (vertical) size. To circumvent this problem, we provide all the three feature maps to RPN; each feature map is assigned to extracting objects within a certain size range. As the semantic information is accumulated at the deeper layers of a convolutional network, feature maps from the first blocks provide a shallow representation of an image that in our case might not be sufficient to filter out background and other complex artifacts. To enrich these first feature maps with more complex representation while preserving their resolution, we implement an architecture similar to the Feature Pyramid Network²⁴. First, three feature maps with sequentially decreasing angular resolution are obtained from the residual blocks as discussed above. After that, smaller and semantically stronger feature maps are summed via upsampling and lateral connections with larger feature maps. To ensure the same number of channels $C = 64$ among the feature maps, the lateral connections are preceded by convolutional layers with 1×1 kernels.

It is worth noting that the model can be applied on images of arbitrary size; the shapes discussed above correspond to the simulated images with fixed size 512×512 (see Section IV C), but for the experimental images the resolution might be increased and it generally depends on the initial resolution of the area detector used in the experiment.

Region proposal network. Our RPN architecture follows the one from²⁵ with a reduced number of channels $C = 64$. It is applied to each of the three feature maps with the corresponding anchors (see Supplementary Table 1).

Unlike for usual objects on RGB photographs, there are no well-defined edges of a Gaussian peak; the radial size of a box is arbitrarily defined as a Gaussian RMS width of a peak. However, the surrounding background and overall profile is essential for correct identification of a peak on an image. To address this issue, we extend the target object boxes for RPN by applying padding $w_{RPN} = 1.1w_{sim} + 1.5$ pixels, so that the box (therefore, the extracted region on the second detection stage) contains the surrounding background.

Second detection stage. To extract a proposed RoI from a feature map, we use RoIAlign layer introduced in²⁶ that reshapes a RoI from a feature map into a fixed-shape tensor via bilinear interpolation. Compared to the basic implementation, we increase this resulting shape from 7×7 to 16×16 to provide a better resolution and, therefore, a better box regression accuracy. At the same time, we decrease the representation size (number of channels) of a RoI from 1024²⁵ to 32.

Unlike usual objects in photographs with complex sharp shapes, several distinct segments at the same $|Q|$ position may be confused with one pronounced segment, especially if a feature map size is drastically reduced along the angular axis. To reduce the probability of this type of detection errors, we only use the first largest feature map for the second detection stage, i.e., RoIAlign layer extracts all the RPN proposals from the largest feature map (see Fig. 1). In this way, the use of Feature Pyramid Network is essential to provide maximum information for the second detection stage.

We reduce the classifier to two classes (object/background) and use sigmoid loss function instead of cross-entropy loss just as in RPN classifier. The resulting score of confidence is assigned to refine an objectness score from RPN.

Simulated data for the training

Training the model on the experimental data would require enormous efforts in annotating large amounts of the measured data manually. Even then there would be a high risk of overfitting on some particular types of data. We find that simulating the data via simple heuristics is sufficient to achieve a very good performance on various types of experimental data.

When designing the simulation process, we aim at training the model to identify vertical lines with 2D-Gaussian profiles on the images superimposed by different experimental artifacts, levels of noise and background signals. An image size of a simulation is 512×512 pixels. Most of the simulation stages described below are optional and are invoked with some probability to cover various combinations of the artifacts and features. Moreover, we define probability distributions for most of the simulation parameters so that they differ for every new simulation. The peak positions are stochastic and model-free; the neural network is trained to detect each of the peaks regardless of the position of the other peaks. In this way, the simulations are not limited to a subset of predefined crystal systems; this approach allows to analyze various complex mixed systems with the same neural network for peak detection.

Each peak profile is modeled by a two-dimensional Gaussian function. To simulate angular profiles of the peaks, which depend on the orientation distribution of the crystalline grains and may exhibit extremely variable patterns, we multiply a simulated map of 2D-Gaussian peaks by Perlin noise⁴⁵. The simulated background consists of several optional components: linear background, Perlin noise, broad Gaussian profiles. A weighted sum of a background and peak profiles map is further modified by applying Poisson noise to simulate counting statistics, adding detector gaps and geometry-dependent dark areas, etc. The simulation process is organized as a series of sequential image processing steps, and each step adds a certain artifact with a defined probability, so in general each image features multiple artifacts and types of background (see Supplementary Fig. 1 and the code for the implementation details). Finally, simulated images are processed in a similar way as the experimental ones by applying histogram equalization followed by normalization to the range $I \in [0, 1]$. Figure 2 shows several examples of the simulated images.

Training

The model was implemented in the Pytorch deep learning framework⁴⁶. We trained both RPN and Fast R-CNN together for 3000 iterations; on each iteration, we simulated a batch of 16 grayscale images with size 512×512 , in total 48,000 images. We used the AdamW optimizer (Adam with decoupled weight decay regularization⁴⁷) and halved the learning rate every 500th iteration starting from $lr = 0.002$. The loss function contains four components [Eq. 2]:

$$L = \lambda_1 L_{reg}^{RPN} + \lambda_2 L_{score}^{RPN} + \lambda_3 L_{reg}^{ROI} + \lambda_4 L_{score}^{ROI}, \quad (2)$$

where *reg* terms are regression losses for box coordinates calculated as smooth L1 loss²⁵, *score* terms are objectness losses calculated as a sigmoid function, *RPN* and *ROI* denote the first and the second detection stages, respectively, and λ_i are the weights used to balance these terms; we use $\lambda_1 = \lambda_3 = 10$ and $\lambda_2 = \lambda_4 = 1$.

The training data is simulated on the fly during the training. In this way, we did not train a model on a same image twice to eliminate possible overfitting, increase variability of the data, and omit a validation set. The whole training takes <25 min on a single NVIDIA 2080Ti graphics card. It is worth noting that, since our model is much lighter (i.e., contains much less parameters) than the original Faster R-CNN implementation, it allows to increase batch size from 2 to 16 images per batch and, therefore, unfreeze and train Batch Normalization layers^{25,48}.

Postprocessing

One disadvantage of the Faster R-CNN architecture is that a single object can be detected several times with overlapping boxes. Thus, after obtaining the box coordinates and scores of confidence, some of the predictions have to be filtered in order to eliminate possible duplicates. In this work, we use a standard non-maximum suppression operation⁴⁹ which removes all overlapping boxes except the one with the highest score of confidence if the degree of overlap exceeds an IoU metric²⁵ $IoU = 0.1$. The value of *IoU* is set quite low to ensure minimal appearance of duplicates, but it can be adjusted if many overlapping peaks are expected on a diffraction pattern.

The other necessary filtering stage that was implemented is the removal of predictions with low confidence score. In this work, we use the threshold value 0.8, though most of the predicted peaks exhibit a confidence score above 0.95. When detecting the peaks in situ data, it is desirable to identify and connect Bragg reflections that appear in sequential time frames. This is achieved by calculating *IoU* between all the detected peaks from adjacent time frames and connecting those pairs of peaks with largest *IoU* if it is sufficient (we use an arbitrarily chosen value

$IoU = 0.3$; the results of the algorithm are not sensitive to this parameter). As a result, we obtain a set of reflections with detected positions over time and total duration of the presence of each peak. Based on this duration, we are able to perform an additional filtering procedure by removing those peaks with small duration as they are likely to be related to experimental artifacts or noise.

Comparison to other object detection models

In addition to our two-stage detector, we trained the other three neural networks and evaluated their performance on the simulated data (see Supplementary Fig. 2).

Faster R-CNN model with ResNet-50 and Feature Pyramid Network²⁴ with two object classes (peaks and background) is trained with four images per batch due to memory limitations. Therefore, the training is extended to 12,000 iterations.

U-Net³³ is modified by adding Batch Normalization layers and trained to solve a segmentation task with a cross-entropy loss; the separate peak positions are obtained from the segmentation map via the standard algorithm⁵⁰. The confidence score of each peak prediction is estimated by the mean value of the segmentation map within the predicted box. Similar to Faster R-CNN, we use four images per batch due to memory limitations and 12,000 training iterations in total.

Our RPN (a one-stage detector) is trained separately; the training process is equivalent to our two-stage detector.

Phase identification and indexing

For a given crystal structure and orientation, we calculate reflection positions and structure factors for all the combinations of miller indices in a wide range $[-20, 20]$, filter them based on $Q_z, Q_{||}$ ranges measured in the experiment, and determine which reflections appear close enough to the detected peaks based on the following conditions: $||Q_{detected} - |Q_{sim}||/w_{detected} \leq 1$ and $|\phi_{detected} - \phi_{sim}|/a_{detected} \leq 1$ where $w_{detected}$, $a_{detected}$ are radial and angular sizes of the detected peak, respectively, and $\phi = \arctan(Q_z/Q_{||})$. The fraction of the corresponding structure factors gives a metric which is sufficient for finding the best match among a set of phases and orientations (see Supplementary Fig. 4). Since some of the reflections appear at $Q_{||} = 0$ ($\phi = 90^\circ$) and their centers are hidden behind a missing wedge, we automatically prolong those peaks that reach the edge of a missing wedge to $\phi = 90^\circ$ to be matched correctly.

The minimal number of detected peaks required to identify the structure correctly depends on the choice of an indexing algorithm. In general, the employed matching algorithm might require substantially more detected peaks to robustly identify the correct structure compared to standard indexing algorithms that may only require three peaks per structure. Nevertheless, the number of peaks detected by the model on the experimental datasets is sufficient to determine every presented structure with the matching algorithm correctly.

When there are both reflections from crystalline grains with preferred orientations (short segments in angular dimension) and long rings from random orientations, it is beneficial to separate the predictions to these two groups and index separately. We do so for the analysis of 2D perovskite data by clustering predicted peaks into two groups based on the relative angular size of peaks $a_{detected}/H(Q_{detected})$, where $H(Q_{detected})$ is the maximum possible angular size at a given $|Q|$ position. In this case, the reflections from ITO and kapton are separated from the reflections from 2D perovskites and they are indexed separately.

The indexing stage is rather trivial and it is performed by assigning the closest simulated reflection to each detected peak given the reflection is close enough (see the conditions above) for each of the phases. Thus, one peak may correspond to several overlapping reflections from different phases (see Fig. 3).

Unit cell refinement

Lattice parameters of the identified $n=2$ and $n=3$ phases of the $(BA)_2(MA)_{n-1}Pb_nI_{3n+1}$ 2D perovskite are refined based on the positions of the detected and indexed Bragg reflections. We achieve this by minimizing the mean squared distance between the detected and simulated peak positions with respect to the lattice parameters. Some of the reflections from $n=2$ and $n=3$ phases overlap and are detected as single peaks. Indeed, the corresponding detected peak positions are, in general, biased. To achieve a better accuracy of the lattice parameters determination, we only use those N peaks that do not overlap with peaks from another phase.

The refined lattice parameters are obtained by minimizing [Eq. 3]

$$\chi_{N-3}^2(a, b, c) = \sum_{i=1}^N \left(\frac{Q_{\text{detected}}^i - Q_{\text{sim}}^i(a, b, c, h_i, k_i, l_i)}{\sigma_{\text{detected}}^i} \right)^2 \quad (3)$$

with respect to the lattice parameters a, b, c for an orthorhombic unit cell (angles $\alpha = \beta = \gamma = \pi/2$) for each time frame via L-BFGS-B algorithm⁵¹; the Miller indices $\{h_i, k_i, l_i\}$ of an i^{th} peak are known from the indexing results. The initial values $a = 8.947 \text{ \AA}$, $b = 39.347 \text{ \AA}$, $c = 8.8589 \text{ \AA}$ for $n = 2$ and $a = 8.928 \text{ \AA}$, $b = 51.959 \text{ \AA}$, $c = 8.878 \text{ \AA}$ for $n = 3$ are taken from the previously reported structures³⁶. The standard errors are calculated using inverse Hessian matrices; detection errors are majorized by $\sigma_{\text{detected}}^i = 0.01 \text{ \AA}^{-1}$.

DATA AVAILABILITY

The GIXD datasets are available from the corresponding author upon reasonable request.

CODE AVAILABILITY

The code for object detection model and the simulations is published on <https://github.com/schreiber-lab/mlgixd>.

Received: 23 November 2021; Accepted: 4 April 2022;

Published online: 03 May 2022

REFERENCES

- Kojima, A., Teshima, K., Shirai, Y. & Miyasaka, T. Organometal halide perovskites as visible-light sensitizers for photovoltaic cells. *J. Am. Chem. Soc.* **131**, 6050–6051 (2009).
- Green, M. A., Ho-Baillie, A. & Snaith, H. J. The emergence of perovskite solar cells. *Nat. Photon.* **8**, 506–514 (2014).
- Sinha, S. K., Sirota, E. B., Garoff, S. & Stanley, H. B. X-ray and neutron scattering from rough surfaces. *Phys. Rev. B* **38**, 2297–2311 (1988).
- Hu, Q. et al. In situ dynamic observations of perovskite crystallisation and microstructure evolution intermediated from [PbI₆]⁴⁻ cage nanoparticles. *Nat. Commun.* **8**, 15688 (2017).
- Chen, A. Z. et al. Origin of vertical orientation in two-dimensional metal halide perovskites and its effect on photovoltaic performance. *Nat. Commun.* **9**, 1336 (2018).
- Liu, Y. et al. Stabilization of highly efficient and stable phase-pure FAPbI₃ perovskite solar cells by molecularly tailored 2d-overlayers. *Angew. Chem. Int. Ed.* **59**, 15688–15694 (2020).
- Zhang, H. et al. Multimodal host–guest complexation for efficient and stable perovskite photovoltaics. *Nat. Commun.* **12**, 3383 (2021).
- Wang, C., Steiner, U. & Sepe, A. Synchrotron big data science. *Small* **14**, 1802291 (2018).
- Greco, A. et al. Fast fitting of reflectivity data of growing thin films using neural networks. *J. Appl. Cryst.* **52**, 1342–1347 (2019).
- Greco, A. et al. Neural network analysis of neutron and x-ray reflectivity data: pathological cases, performance and perspectives. *Mach. Learn.: Sci. Technol.* **2**, 045003 (2021).
- Samarakoon, A. M. et al. Machine-learning-assisted insight into spin ice dy₂ti₂o₇. *Nat. Commun.* **11**, 892 (2020).
- Sanchez-Gonzalez, A. et al. Accurate prediction of x-ray pulse properties from a free-electron laser using machine learning. *Nat. Commun.* **8**, 15461 (2017).
- Ziletti, A., Kumar, D., Scheffler, M. & Ghiringhelli, L. M. Insightful classification of crystal structures using deep learning. *Nat. Commun.* **9**, 2775 (2018).
- Ryan, K., Lengyel, J. & Shatruk, M. Crystal structure prediction via deep learning. *J. Am. Chem. Soc.* **140**, 10158–10168 (2018).
- Oviedo, F. et al. Fast and interpretable classification of small x-ray diffraction datasets using data augmentation and deep neural networks. *npj Comput. Mater.* **5**, 60 (2019).
- Tatlier, M. Artificial neural network methods for the prediction of framework crystal structures of zeolites from xrd data. *Neural Comput. Applic.* **20**, 365–371 (2011).
- Lee, J.-W., Park, W. B., Lee, J. H., Singh, S. P. & Sohn, K.-S. A deep-learning technique for phase identification in multiphase inorganic compounds using synthetic XRD powder patterns. *Nat. Commun.* **11**, 86 (2020).
- Wang, B., Yager, K., Yu, D. & Hoai, M. X-ray scattering image classification using deep learning. In *2017 IEEE Winter Conf. Appl. Comput. Vis.*, 697–704 (IEEE, 2017).
- Ke, T.-W. et al. A convolutional neural network-based screening tool for x-ray serial crystallography. *J. Synchrotron Radiat.* **25**, 655–670 (2018).
- Liu, S. et al. Convolutional neural networks for grazing incidence x-ray scattering patterns: thin film structure identification. *MRS Commun.* **9**, 586–592 (2019).
- Sullivan, B. et al. BraggNet: integrating bragg peaks using neural networks. *J. Appl. Crystallogr.* **52**, 854–863 (2019).
- Liu, Z. et al. BraggNN: fast X-ray Bragg peak analysis using deep learning. *IUCr J*, 104–113 (2022).
- Ren, S., He, K., Girshick, R. & Sun, J. Faster R-CNN: towards real-time object detection with region proposal networks. In *Adv. Neural Inf. Process. Syst.*, vol. 28, 91–99 (Curran Associates, Inc., 2015). <https://proceedings.neurips.cc/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf>.
- Lin, T.-Y. et al. Feature pyramid networks for object detection. In *2017 IEEE Conf. Comput. Vis. Pattern Recogn.*, 936–944 (IEEE, 2017). <https://doi.org/10.1109/cvpr.2017.106>.
- Girshick, R. Fast R-CNN. In *2015 IEEE Int. Conf. Comput. Vis.*, 1440–1448 (IEEE, 2015).
- He, K., Gkioxari, G., Dollár, P. & Girshick, R. Mask R-CNN. In *2017 IEEE Int. Conf. Comput. Vis.*, 2961–2969 (IEEE, 2017).
- Redmon, J., Divvala, S., Girshick, R. & Farhadi, A. You only look once: Unified, real-time object detection. In *2016 IEEE Conf. Comput. Vis. Pattern Recogn.*, 779–788 (IEEE, 2016).
- Carion, N. et al. End-to-end object detection with transformers. In *Comput. Vis. ECCV*, 213–229 (Springer, 2020).
- Rietveld, H. M. Line profiles of neutron powder-diffraction peaks for structure refinement. *Acta Cryst.* **22**, 151–152 (1967).
- Rietveld, H. M. A profile refinement method for nuclear and magnetic structures. *J. Appl. Crystallogr.* **2**, 65–71 (1969).
- David, W. I. F. Powder diffraction peak shapes. parameterization of the pseudo-voigt as a voigt function. *J. Appl. Crystallogr.* **19**, 63–64 (1986).
- Dinapoli, R. et al. Eiger: Next generation single photon counting detector for x-ray applications. *Nucl. Instrum. Methods Phys. Res. A* **650**, 79–83 (2011).
- Ronneberger, O., Fischer, P. & Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Lecture Notes in Computer Science*, 234–241 (Springer International Publishing, 2015). https://doi.org/10.1007/978-3-319-24574-4_28.
- Wang, C.-Y., Yeh, I.-H. & Liao, H.-Y. M. You only learn one representation: Unified network for multiple tasks. *Preprint at https://arxiv.org/abs/2105.04206* (2021).
- Stoumpos, C. C., Malliakas, C. D. & Kanatzidis, M. G. Semiconducting tin and lead iodide perovskites with organic cations: Phase transitions, high mobilities, and near-infrared photoluminescent properties. *Inorg. Chem.* **52**, 9019–9038 (2013).
- Stoumpos, C. C. et al. Ruddlesden–popper hybrid lead iodide perovskite 2d homologous semiconductors. *Chem. Mater.* **28**, 2852–2867 (2016).
- Soe, C. M. M. et al. Structural and thermodynamic limits of layer thickness in 2d halide perovskites. *PNAS* **116**, 58–66 (2018).
- Seeck, O. H. et al. The high-resolution diffraction beamline P08 at PETRA III. *J. Synchrotron Rad.* **19**, 30–38 (2012).
- Newville, M., Stensitzki, T., Allen, D. B. & Ingarziola, A. Lmfit: non-linear least-square minimization and curve-fitting for python <https://zenodo.org/record/11813> (2014).
- Huang, X. et al. Interactive visual study of multiple attributes learning model of x-ray scattering images. *IEEE Trans. Vis. Comput. Graph.* **27**, 1312–1321 (2021).
- Jha, D. et al. Peak area detection network for directly learning phase regions from raw x-ray diffraction patterns. In *2019 Proc. Int. Conf. Neural Netw.* (IEEE, 2019). <https://doi.org/10.1109/ijcnn.2019.8852096>.
- Baker, J. L. et al. Quantification of thin film crystallographic orientation using x-ray diffraction with an area detector. *Langmuir* **26**, 9146–9151 (2010).
- Stark, J. A. *Adaptive image contrast enhancement using generalizations of histogram equalization*. vol. 9, 889–896 (IEEE, 2000).
- He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recogn.*, 770–778 (IEEE, 2016).
- Perlin, K. An image synthesizer. *SIGGRAPH Comput. Graph.* **19**, 287–296 (1985).
- Paszke, A. et al. Pytorch: an imperative style, high-performance deep learning library. In *Adv. Neural Inf. Process. Syst.*, vol. 32 (Curran Associates, Inc., 2019). <https://proceedings.neurips.cc/paper/2019/file/bd8c288fee7f92f2bfa9f701272740-Paper.pdf>.
- Loshchilov, I. & Hutter, F. Decoupled weight decay regularization. In *ICLR* <https://openreview.net/forum?id=Bkg6RiCqY7> (2019).
- Ioffe, S. & Szegedy, C. Batch normalization: accelerating deep network training by reducing internal covariate shift. In *Proc. 32nd Int. Conf. Mach. Learn. ICML 2015*, vol. 37 of *Proc. Mach. Learn. Res.*, 448–456 (PMLR, Lille, France, 2015). <https://proceedings.mlr.press/v37/ioffe15.html>.
- Neubeck, A. & Van Gool, L. Efficient non-maximum suppression. In *Proc. Int. Conf. Pattern Recognit.*, vol. 3, 850–855 (IEEE, 2006).

50. Wu, K., Otoo, E. & Shoshani, A. Optimizing connected component labeling algorithms. In *SPIE Proc.* (SPIE, 2005). <https://doi.org/10.1117/12.596105>.
51. Byrd, R. H., Lu, P., Nocedal, J. & Zhu, C. A limited memory algorithm for bound constrained optimization. *SISC* **16**, 1190–1208 (1995).

ACKNOWLEDGEMENTS

This research is part of a project funded by the German Federal Ministry for Science and Education (BMBF).

We acknowledge DESY (Hamburg, Germany), a member of the Helmholtz Association HGF, for the provision of experimental facilities. Parts of this research were carried out at PETRA III and we would like to thank Chen Shen and Rene Kirchof for assistance in using photon beamline P08. Beamtime was allocated for proposal II-20190761. This research was also supported in part through the Maxwell computational resources operated at DESY with the assistance of André Rothkirch and Frank Schlünzen.

We acknowledge Laboratory of Photonics and Interfaces at école Polytechnique Fédérale de Lausanne and the Adolphe Merkle Institute of the University of Fribourg and gratefully thank Jovana Milić, Anwar Alanazi and Algirdas Ducinkas for providing the samples.

We thank the Deutsche Forschungsgemeinschaft (DFG) for financial support.

Supported by the German Research Foundation through the Cluster of Excellence “Machine Learning—New Perspectives for Science”. Frank Schreiber is a member of the Machine Learning Cluster of Excellence, funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy - EXC number 2064/1 - Project number 390727645.

AUTHOR CONTRIBUTIONS

V.S. and A.H. conceived the concept for the work. V.S. wrote the code, designed and trained the neural network. V.M. carried out the conventional analysis of GIXD data and took part in designing the simulations. F.B., E.K., V.M., and A.H. carried out the measurements. V.S., V.M., A.Gr., A.P., A.G., and F.S. wrote the paper and contributed to the analysis and interpretation of the results. F.S. supervised the research process. All authors discussed the results and commented on the paper.

FUNDING

Open Access funding enabled and organized by Projekt DEAL.

COMPETING INTERESTS

The authors declare no competing interests.

ADDITIONAL INFORMATION

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s41524-022-00778-8>.

Correspondence and requests for materials should be addressed to Vladimir Starostin, Alexander Hinderhofer or Frank Schreiber.

Reprints and permission information is available at <http://www.nature.com/reprints>

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2022