

Robust recognition and exploratory analysis of crystal structures via Bayesian deep learning

Andreas Leitherer^{1,*}, Angelo Ziletti¹, and Luca M. Ghiringhelli¹

¹Fritz-Haber-Institut der Max-Planck-Gesellschaft, 14195 Berlin-Dahlem, Germany

*leitherer@fhi-berlin.mpg.de

ABSTRACT

Due to their ability to recognize complex patterns, neural networks can drive a paradigm shift in the analysis of materials science data. Here, we introduce ARISE, a crystal-structure identification method based on Bayesian deep learning. As a major step forward, ARISE is robust to structural noise and can treat more than 100 crystal structures, a number that can be extended on demand. While being trained on ideal structures only, ARISE correctly characterizes strongly perturbed single- and polycrystalline systems, from both synthetic and experimental resources. The probabilistic nature of the Bayesian-deep-learning model allows to obtain principled uncertainty estimates, which are found to be correlated with crystalline order of metallic nanoparticles in electron tomography experiments. Applying unsupervised learning to the internal neural-network representations reveals grain boundaries and (unapparent) structural regions sharing easily interpretable geometrical properties. This work enables the hitherto hindered analysis of noisy atomic structural data from computations or experiments.

The goal of crystal-structure identification is to assign a symmetry label (for instance, the space group) to a given atomic structure. More generally, one wants to find the most similar structure within a list of given known systems – the *structural classes*, identified by stoichiometry, space group, number of atoms in the unit cell, and location of the atoms in the unit cell (the Wyckoff positions). This can help to understand important physical properties such as the hardness of industrial steel¹. In this context, the importance of grain boundaries has been studied in numerous experiments², also in combination with crystal-structure prediction³. Beyond bulk materials, two- (2D) and one-dimensional (1D) systems have far-reaching technological applications, such as solar energy storage, DNA sequencing, cancer therapy, or even space exploration^{4,5}.

Methods for automatic crystal-structure recognition are required to analyze the continuously growing amount of geometrical information on crystal structures, from both experimental and computational studies. Millions of crystal structures alongside calculated properties are available in large computational databases such as the Novel Materials Discovery (NOMAD) Laboratory^{6,7}, AFLOW^{8,9}, the Open Quantum Materials Database (OQMD)¹⁰, Materials Project¹¹, or repositories specialized in 2D materials^{12,13}. In scanning transmission electron microscopy (STEM)¹⁴, atomic positions can be reconstructed from atomic-resolution images for specific systems, e.g., graphene¹⁵. Three-dimensional atomic positions are provided by atom probe tomography (APT)¹⁶ and atomic electron tomography (AET) experiments¹⁷. Still, substantial levels of noise due to experimental limitations and reconstruction errors are present in atomic positions, e.g. distortions beyond the physical level or, in case of APT, large amount of missing atoms (at least 20%, due to the limited detector efficiency¹⁸). Crystal-structure recognition schemes should be able to classify a large number of structural classes (also be-

yond bulk materials) while at the same time being robust with respect to theoretical or experimental sources of inaccuracy and physically driven deviations from ideal crystal symmetry (e.g., vacancies or thermal vibrations). Given the large amount of data, the classification should be fully automatic and independent of the manual selection of tolerance parameters (which quantify the deviation from an ideal reference structure). Current methods are based either on space-group symmetry or local structure. For space-group-based approaches (notable examples being Spglib¹⁹ and AFLOW-SYM²⁰), the allowed symmetry operations are calculated directly from the atomic positions to infer a space group label. For local-structure-based approaches, the local atomic neighborhood of each individual atom is classified into a predefined list of reference structures. Examples of these methods are common neighbor analysis (CNA)²¹, adaptive common neighbor analysis (a-CNA)²², bond angle analysis (BAA)²³, and polyhedral template matching (PTM)²⁴. Space-group approaches can treat all space groups but are sensitive to noise, while local-structure methods can be quite robust but only treat a handful of structural classes. Moreover, none of the available structure recognition schemes can recognize complex nanostructures, e.g., nanotubes.

To improve on the current state of the art, we build on recent advances in deep learning, which is a subfield of machine learning that yields ground-breaking results in many settings, e.g., image and speech recognition²⁵. Previous work using machine learning and neural networks (NNs) for crystal-structure recognition^{26–29} did not go beyond a handful of structural classes while showing robustness at the same time. Here, we propose a robust, threshold-independent crystal-structure recognition framework (Artificial-Intelligence-based Structure Evaluation, short ARISE) to classify a diverse set of 108 structural classes, comprising bulk, 2D, and 1D materials. Bayesian NNs^{30,31} are used, i.e., a recently developed family

of NNs that yields not only a classification but also uncertainty estimates. These estimates are principled in the sense that they approximate those of a well-known probabilistic model (the Gaussian process). This allows to quantify prediction uncertainty, but also the degree of crystalline order in a material. ARISE performance is compared with the current state of the art, and then applied to various computational and experimental atomic structures. Crystal characterization and identification of hidden patterns is performed using supervised learning (ARISE) as well as the unsupervised analysis (via clustering and dimensionality reduction) of the internal representations of ARISE.

Results

The input representation

To apply machine learning to condensed-matter and materials science problems, the input coordinates, chemical species, and the lattice periodicity of a given atomic structure are mapped onto a suitable *descriptor*. Here, the descriptor is a vector that is invariant under rigid translations and rotations of the input structure, as well as under permutations of same-species atoms. Quality and generalization ability of machine-learning models can be significantly increased, if physical requirements known to be true are respected by construction (see section S1.2 for more details).

Most well-known descriptors in physics and materials science incorporate these physical invariants: symmetry functions³², the smooth-overlap-of-atomic-positions descriptor (SOAP)^{33,34}, the many-body tensor representation³⁵, and the moment tensor potential representation³⁶. In this work, SOAP is used as descriptor (cf. section S1.2). SOAP has been successfully applied to numerous materials science problems such as interatomic potentials fitting³⁷, structural similarity quantification³⁸, or prediction of grain boundary characteristics (e.g., energy and mobility)³⁹. Note that any other suitable descriptor that respects above-mentioned physical requirements can be used as input for our procedure. In particular, the *aiMaterials* code library is provided into which alternative descriptors can be readily integrated.

The Bayesian deep learning model and the training dataset

Once the crystal structures are converted into vectorial descriptors by means of the SOAP mapping, a NN model is used to arrive at a classification decision (cf. Fig. 1c). NNs are non-linear classifiers: they transform the input in a hierarchical fashion by subsequently applying affine and non-linear transformations in a predefined series of layers. The NN learns these optimal transformations that deform the descriptor space so that a robust classification is achieved. In this way, the model is able to learn complex representations which are becoming more abstract from layer to layer²⁹. This ability to learn representations⁴⁰ is one of the key characteristics distinguishing NNs from other machine-learning algorithms. Various NN architectures have been developed in recent years²⁵,

in this work, a fully-connected NN (multilayer perceptron) is employed.

A key component of this work is something rarely addressed in machine learning applied to materials science: quantification of model prediction uncertainty (cf. Fig. 1d). Standard NNs are unable to provide reliable model classification uncertainty³⁰. There is a widespread use of the probability provided by the last layer as uncertainty estimate. Typically, the so-called softmax function is employed to normalize the sum of output values such that they can be interpreted as classification probabilities. The class with maximal probability corresponds to the final prediction (here of a specific structural class). One may interpret the classification probability as quantification of model confidence. However, this strategy is unreliable as standard NNs tend to erroneously assign unjustified high confidence to points for which a low confidence should be returned instead³⁰. The main reason for this behavior is that standard-NN predictions are deterministic, with the softmax output only providing point estimates of the true probability distribution of outputs. In Bayesian NNs, this is addressed by placing distributions over model parameters. This results in probabilistic outputs – in contrast to the point estimates from deterministic NNs – from which principled uncertainty estimates can be obtained. Gal and Ghahramani³⁰ showed that high-quality uncertainty estimates (alongside predictions) can be calculated at low cost using stochastic regularization techniques such as dropout^{41,42} (see section S1.3 for more details).

After both descriptor and model architecture have been identified, a diverse, comprehensive, and materials-science-relevant training set is constructed. The first – and most important – step is to define the structural classes which are going to be included in the model: an overview of the structural classes considered in this work is shown in Fig. 1e. This comprehensive collection of structures includes bulk materials of elemental, binary, ternary, and quaternary composition, as well as 2D materials and carbon nanotubes of chiral, arm-chair, and zigzag type. In practice, given any database, we extract prototypes, i.e., representative structures that are selected according to some predefined rules. Selection criteria are, for instance, fulfillment of geometrical constraints (number of atoms in the unit cell, number of chemical species) or if the structures are observed in experiment. For the elemental bulk materials, we extract from AFLOW all experimentally observed structures with up to four atoms in the primitive cell. This yields 27 elemental solids encompassing all Bravais lattices, with the exception of monoclinic and triclinic structures because of their low symmetry. Note that this selection includes not only the most common structures such as face-centered-cubic (fcc), body-centered-cubic (bcc), hexagonal-close-packed (hcp), and diamond (which cover more than 80% of the elemental solids found in nature⁴³), but also double-hexagonal close-packed, graphite (hexagonal, rhombohedral, buckled), and orthorhombic systems such as black phosphorus. This goes already beyond previous work using NNs for

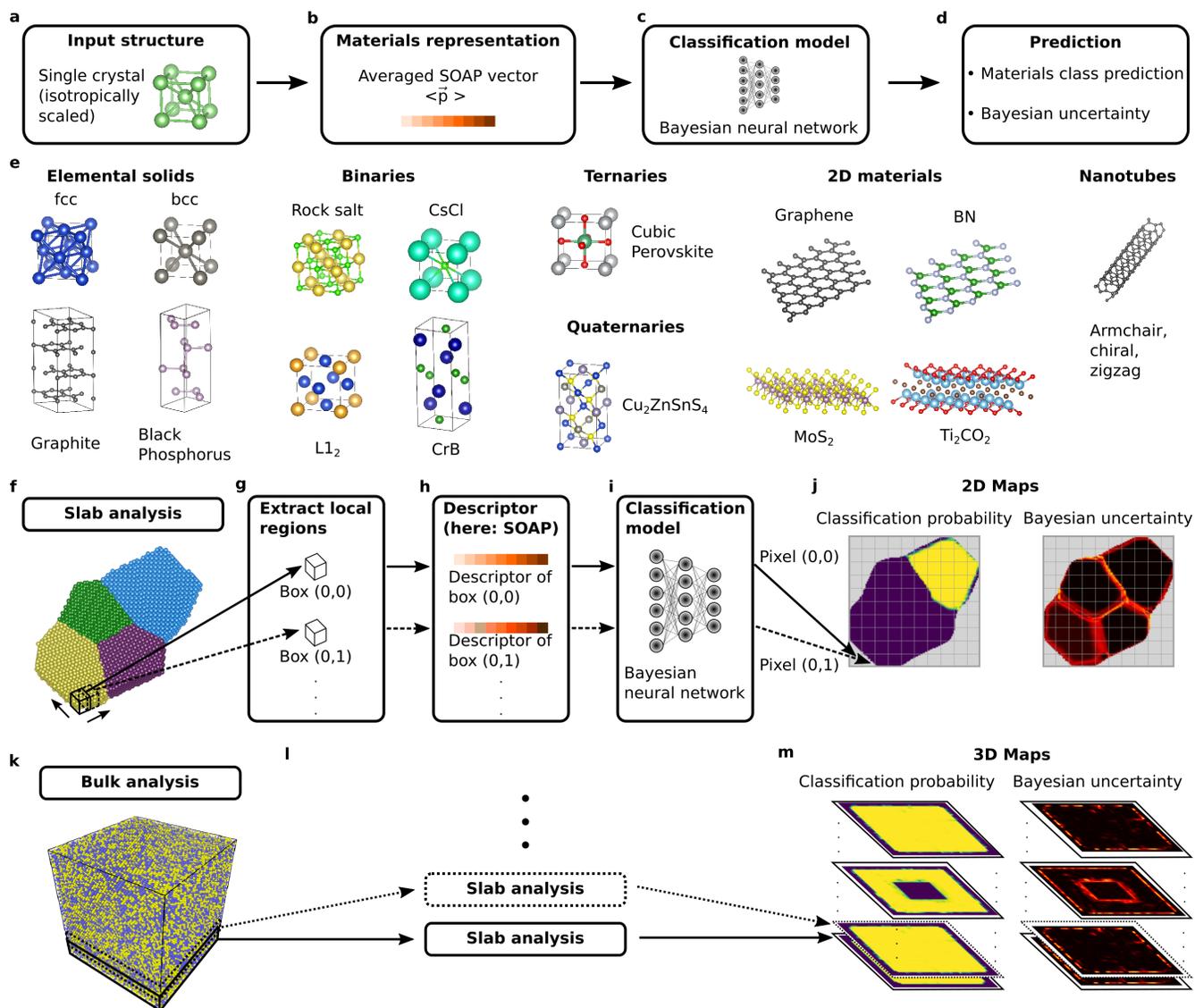


Figure 1. Schematic overview of single- and polycrystal characterization framework. **a-d** Prediction pipeline of single-crystal classification model ARISE. **e** Examples of crystallographic prototypes included in the training set. **f-m** polycrystal classification framework *strided pattern matching* for slab-like (**f-j**) and bulk systems (**k-m**).

crystal structure recognition²⁹, where a smaller set of elemental solids is considered. For binaries, we select the ten most common binary compounds according to Pettifor⁴⁴, plus the L1₂ structure because of its technological relevance – for instance, it being the crystal structure of common precipitates in Ni-based superalloys⁴⁵. This selection also includes non-centrosymmetric structures, i.e. structures without inversion symmetry, such as wurtzite. To challenge the classification method with an increasing number of chemical species, a small set of ternary and quaternary materials is included as a proof-of-concept. Specifically, six ternary perovskites⁴⁶ (organometal halide cubic and layered perovskites) and six quaternary chalcogenides of A₂BCX₄ type⁴⁷ are included due to their relevance in solar cells and photo-electrochemical water splitting devices, respectively. Going beyond bulk materials, we add an exhaustive set of 46 2D materials, comprising not only the well-known elemental structures such as graphene and phosphorene⁴⁸ but also binary semiconductors and insulators (BN, GaN), transition metal dichalcogenides (MoS₂), and one example of metal-organic perovskites with six different chemical species. Ternary, quaternary, and 2D materials are taken from the computational materials repository (CMR)⁴⁹. To demonstrate the ability of the proposed framework to deal with complex nanostructures, 12 nanotubes of armchair, chiral and zigzag type are included in the dataset. For each prototype described above, we calculate the SOAP vector with different parameter settings (see section S1.2 for more details) as well as periodic and non-periodic boundary conditions to have a comprehensive dataset to train a robust classification model. This results in 39204 (pristine) structures included in the training set.

To optimize the model, the set of pristine structures is split, with 80% being used for training and the remaining 20% for validation. For hyperparameter tuning, we employ Bayesian optimization⁵⁰, which allows to optimize functions whose evaluation is computationally costly, making it particularly attractive for deep-learning models. Here, hyperparameters such as learning rate or number of layers are optimized in an automatic, reproducible, and computationally efficient manner to minimize the validation accuracy. A list of candidate models is then obtained, from which the optimal model is selected (see Methods section). We term this model ARISE, and report its architecture in Table 2.

Benchmarking

We now compare ARISE’s performance on pristine and defective structures with state-of-the-art crystal-structure recognition methods, specifically spglib, CNA, a-CNA, BAA, and PTM (cf. Table 1). As mentioned in the Introduction, none of the benchmarking methods can treat all the materials shown in Fig. 1e; thus for fairness, the classification accuracy is only calculated for classes for which the respective methods were designed for, implying that most structures are excluded (see S2.1 for more details).

The performance on pristine structures is reported in Table

1. The accuracy in classifying pristine structures is always 100% as expected, with the only exception being CNA: For this method, the default cutoff only allows to correctly classify fcc and bcc but not hcp structures. For defective structures, the situation is drastically different. Spglib classification accuracy on displaced structures is low, and only slightly improved by using loose setting (up to 1% displacement). For missing atoms, the accuracy is very low already at the 1% level regardless of the setting used. Note, however, this is actually spglib’s desired behavior since the aim of this method is not robust classification. As indicated in the first column of Table 1, spglib can treat 96 out of the 108 prototypes included in our dataset with the twelve missing prototypes being carbon nanotubes. Methods based on local atomic environment (PTM, BAA, CNA, a-CNA) perform very well on displaced structures, but they suffer from a substantial accuracy drop for missing-atoms ratios beyond 1%. Their biggest drawback, however, is that they can treat only a handful of classes: three classes for BAA, CNA, and a-CNA, and twelve classes for PTM. ARISE is very robust with respect to both displacements and missing atoms (even concurrently, cf. Table S3), while being the only method able to treat all 108 classes included in the dataset, including complex systems, such as carbon nanotubes. An uncertainty value quantifying model confidence is also returned, which is particularly important when investigating defective structures or inputs that are far out of the training set. We provide a detailed study in section S2.3, where we challenge ARISE with structures it has not been trained on, i.e., it is forced to fail by construction. We find that ARISE returns non-trivial physically meaningful predictions, thus making it particularly attractive, e.g., for screening large and structurally diverse databases. Moreover, we analyze predictions and uncertainty of ARISE for continuous structural transformations in S2.2, where we consider the so-called Bain path that includes transitions between fcc, bcc, and tetragonal structures.

Polycrystal classification

Up to this point, we have discussed only the analysis of single-crystal (mono-crystalline) structures, using ARISE. To enable the local characterization of polycrystalline systems, we introduce *strided pattern matching* (SPM). For slab-like systems (cf. Fig. 1f), a box of predefined size is scanned in-plane across the whole crystal with a given stride; at each step, the atomic structure contained in the box is represented using a suitable descriptor (cf. Fig. 1g-h), and classified (Fig. 1i), yielding a collection of classification probabilities (here: 108) with associated uncertainties. These are arranged in 2D maps (Fig. 1j). The classification probability maps indicate how much a given polycrystalline structure locally resembles a specific crystallographic prototype. The uncertainty maps quantify the statistics of the output probability distribution (cf. section S1.3). Increased uncertainty indicates that the corresponding local segment(s) deviates from the prototypes known to the model. Thus, these regions are likely to contain

	Pristine	Random displacements (δ)					Missing atoms (η)			
		0.1%	0.6%	1%	2%	4%	1%	5%	10%	20%
Spplib (loose, 96 / 108)	100.00	100.00	100.00	95.26	20.00	0.00	11.23	0.00	0.00	0.00
Spplib (tight, 96 / 108)	100.00	0.00	0.00	0.00	0.00	0.00	11.23	0.00	0.00	0.00
PTM (12 / 108)	100.00	100.00	100.00	100.00	100.00	100.00	88.67	51.76	25.93	6.24
CNA (3 / 108)	66.14	62.81	62.81	54.55	32.34	31.41	55.86	32.50	15.75	3.07
a-CNA (3 / 108)	100.00	100.00	100.00	100.00	100.00	100.00	89.25	52.81	25.92	5.37
BAA (3 / 108)	100.00	100.00	100.00	100.00	100.00	97.85	99.71	88.78	65.21	25.38
ARISE (108 / 108)	100.00	100.00	100.00	100.00	99.86	99.29	100.00	100.00	100.00	99.85

Table 1. Accuracy in identifying the parent class of defective crystal structures. The defective structures are generated by randomly displacing atoms according to a uniform distribution on an interval $[-\delta \cdot d_{\text{NN}}, +\delta \cdot d_{\text{NN}}]$ proportional to the nearest neighbor distance d_{NN} (central panel), or removing $\eta\%$ of the atoms (right panel). The accuracy values shown are in percentage. For benchmarking we use Spplib¹⁹ (with two settings for the precision parameters, “loose” (position/angle tolerance $0.1\text{\AA}/5^\circ$) and “tight” (position/angle tolerance $10^{-4}/1^\circ$), polyhedral template matching (PTM)²⁴, common neighbor analysis (CNA)²¹, adaptive common neighbor analysis (a-CNA)²², and bond angle analysis (BAA)²³. The number of classes which can be treated out of the materials pool in Fig. 1e is shown in parentheses for each method. spglib can assign a space group to all materials except the 12 nanotubes. PTM can only classify 7 elemental and 5 binary materials of those considered in this work. Additional classes are missing for CNA, a-CNA, and BAA as they cannot classify simple cubic (sc) and diamond structures. The approach proposed here can be applied to all classes, and thus the whole dataset is used (see Tables S4-S6 for a complete list).

defects such as grain boundaries, or more generally atomic arrangements different from the ones included in training. For bulk systems (Fig. 1k), the slab analysis depicted in Fig. 1f-j is repeated for multiple slices (Fig. 1l), resulting in 3D classification probability and uncertainty maps (Fig. 1m).

SPM extends common approaches such as labeling individual atoms with symmetry labels²², as the striding allows to discover structural transitions within polycrystals in a smooth way. SPM can be applied to any kind of data providing atomic positions and chemical species. Results obtained via SPM are influenced by the quality of the classification model as well as box size and stride (see Methods section for more details).

Synthetic polycrystals

First, the classification via SPM is performed for a slab-like synthetic polycrystal consisting of fcc, bcc, hcp, and diamond grains (cf. Fig. 2a). Due to the nature of the system, the SPM boxes near the grain boundaries will contain mixtures of different crystal structures. The results are shown in Fig. 2b and c: The network assigns high classification probability to the correct prototypes. Uncertainty is low within the grains, increasing at grain boundaries and crystal outer borders in line with physical intuition. The result remains virtually unchanged when introducing atomic displacements (up to 1% of the nearest neighbor distance) while concurrently removing 20% of the atoms (cf. Fig. S4). The highest classification probabilities (after from the top four shown in Fig. 2b) are shown in Fig. S7; a discussion on the stride can be found in Fig. S8.

Going beyond classification, we show how unsupervised learning can be used to access structural similarity information embedded in ARISE’s internal representations, and use it for the characterization of crystal systems. We consider

the mono-species polycrystal shown in Fig. 2a and collect ARISE’s representations of the overall 7968 local boxes. Next, we employ Hierarchical Density-based Spatial Clustering Applications with Noise (HDBSCAN)^{51,52} to identify clusters in the high-dimensional representation space. HDBSCAN estimates the density underlying a given data set and then constructs a hierarchy of clusters, from which the final clustering can be obtained via an intuitive and tunable parameter (see Methods). The obtained clusters correspond to the four crystalline grains in the structure (Fig. 2d). Points identified as outliers (marked in orange) coincide with grain-boundary and outer-border regions. Next, the high-dimensional manifold of the NN representations is projected in 2D via Uniform Manifold Approximation and Projection (UMAP)⁵³. UMAP models the manifold underlying a given dataset and then finds a low-dimensional projection that can capture both global and local distances of the original high-dimensional data. This returns a structure-similarity map (Fig. 2e), which allows to visually investigate similarities among structures: points (structures) close to each other in this map are considered to be similar by the algorithm. Structures belonging to the same cluster are in close proximity to each other, and clearly separated from other clusters. Conversely, outlier points are split across different regions of the map. This is physically meaningful: outliers are not a cohesive cluster of similar structures, but rather comprise different types of grain boundaries (i.e., fcc to bcc transitions or fcc to diamond etc., cf. Fig. S9). In this synthetic setting, we can also use the classification prediction to further verify the unsupervised analysis: the results obtained via unsupervised learning indeed match ARISE’s predictions (cf. Fig. 2e - Fig 2f). Moreover, an analysis of the mutual information (Fig. 2g) reveals that points at the core of the clusters are associated with low uncertainty, while points

closer to the boundaries show increased uncertainty. Similar results are obtained for the other layers (cf. Fig. S6).

We now move to a more realistic system: a model structure for Ni-based superalloys⁴⁵ (c.f. Fig. 2h). Ni-based superalloys are used in aircraft engines due to their large mechanical strength at high temperatures, which derives from ordered L1₂ precipitates (γ' phase) embedded in a fcc matrix (γ phase). We generate an atomic structure consisting of a fcc matrix in which Al and Ni atoms are randomly distributed. In the center, however, the arrangement of Al and Ni atoms is no longer random, but it is ordered such that the L1₂ phase is created (c.f. Fig. 2h). The cubic shape of this precipitate is in accordance with experimental observations⁵⁴. The resulting structure comprises 132 127 atoms over a cube of 120 Å length. As shown via a section through the center in Fig. 2i, fcc is correctly assigned to the matrix, and the precipitate is also detected. The uncertainty is increased at the boundary between random matrix and precipitate, as well as at the outer borders. Fig. 2j illustrates the L1₂ classification probability in a 3D plot. The precipitate is detected in both pristine and highly-defective structures. This analysis demonstrates that ARISE can distinguish between chemically ordered and chemically disordered structures, a feature that will be exploited for the analysis of experimental data in Sec. “Application to atomic electron tomography data”.

Application to STEM experimental images

We now investigate defective structures originating from a completely different data source, namely scanning transmission electron microscopy (STEM) experiments, to demonstrate the generalization ability of ARISE and its applicability to experimental data. Moreover, we show how global and local analysis can be combined to analyze crystal structures. STEM experiments are a valuable resource to characterize material specimens, and to study, for instance, the atomic structures at grain boundaries³. Atomic resolution can be achieved in high-angle dark annular field (HAADF) images. The global assignments of ARISE are tested on two experimental HAADF images of graphene shown in Fig. 3a. These images contain a substantial amount of noise which makes it very challenging to recognize the graphene honeycomb pattern by naked eye. The choice of graphene is motivated by it being a flat 2D material; x and y atomic positions obtained from STEM images thus provide the actual crystal structure, and not a mere projection. Approximate atomic positions (i.e. x and y coordinates) from HAADF images are obtained via AtomNet¹⁵, and shown in Fig. 3b. ARISE is then used to classify the structures following the steps summarized in Fig. 1a-d. The top predictions ranked by classification probability are shown in Fig. 3c, together with the uncertainty of the assignments as quantified by the mutual information. ARISE correctly recognizes both images as graphene, despite the substantial amount of noise present in images and reconstructed atomic positions. For the first image (Fig. 3a, left), graphene is predicted with very high probability ($\sim 99\%$). In-

deed, the similarity to graphene is apparent, although evident distortions are present in some regions (e.g. misaligned bonds marked in Fig. 3b). The second candidate structure is C₃N, predicted with $\sim 1\%$ probability; in C₃N, atoms are arranged in a honeycomb lattice, making also this low probability assignment physically meaningful. For the second image (Fig. 3a, right), ARISE also correctly predicts graphene, this time with 79% probability. The uncertainty is six times larger than in the previous case. Indeed, this structure is much more defective than the previous one: it contains a grain boundary in the lower part, causing evident deviations from the pristine graphene lattice, as illustrated in Fig. 3b (right). The other four candidate structures appearing in the top five predictions (PbSe, MnS₂, BN, C₃N) are the remaining completely flat monolayers known to the network (out of the 108 structures in the training dataset, only five are flat monolayers). Note that no explicit information about the dimensionality of the material is given to the model. It is also important to point out that ARISE robustness well beyond physical levels of noise is essential to achieve the correct classification despite the presence of substantial amount of noise from both experiment and atomic position reconstruction.

While so far we have analyzed HAADF images on a global scale, a local analysis via SPM allows to zoom into a given structure and locate sub-structural features. This is particularly useful for polycrystalline and/or larger systems (e.g., more than 1 000 atoms). As illustrative example, we consider the structure in Fig. 3e. The mutual information shown in Fig. 3f (right) clearly reveals the presence of a grain boundary. In Fig. 3 (left), the classification probabilities of graphene and MnS₂ (the dominant prototypes) are presented, the latter being assigned at the grain boundary. This assignment can be traced back to pentagon-like patterns appearing near the grain boundary (as highlighted in Fig. 3f), a pattern similar to the one being formed by Mn and S atoms in MnS₂ (cf. Fig. 3e).

Application to atomic electron tomography data

While HAADF images are a valuable experimental resource, they only provide 2D projections. 3D structural and chemical information can however be obtained from atomic electron tomography (AET) with atomic resolution achieved in recent experiments^{17,55–57}. Notably, this technique provides 3D atomic positions labeled by chemical species, to which ARISE and SPM can be readily applied. While extensions to other systems such as 2D materials are reported⁵⁸, metallic nanoparticles have been the main experimental focus so far, specifically FePt systems due to their promises for biomedicine and magnetic data storage⁵⁹. First, a FePt nanoparticle⁶⁰ is classified using SPM. ARISE’s robustness is critical for this application, since the structural information provided by AET experiments are based on reconstruction algorithms that causes visible distortions (cf. Fig. 4a). SPM primarily detects L1₂, L1₀, and fcc phases (see Fig. S10). This is in line with physical expectations: annealing leads to structural transitions from chemically disordered to ordered fcc (A1 to L1₂) or to the

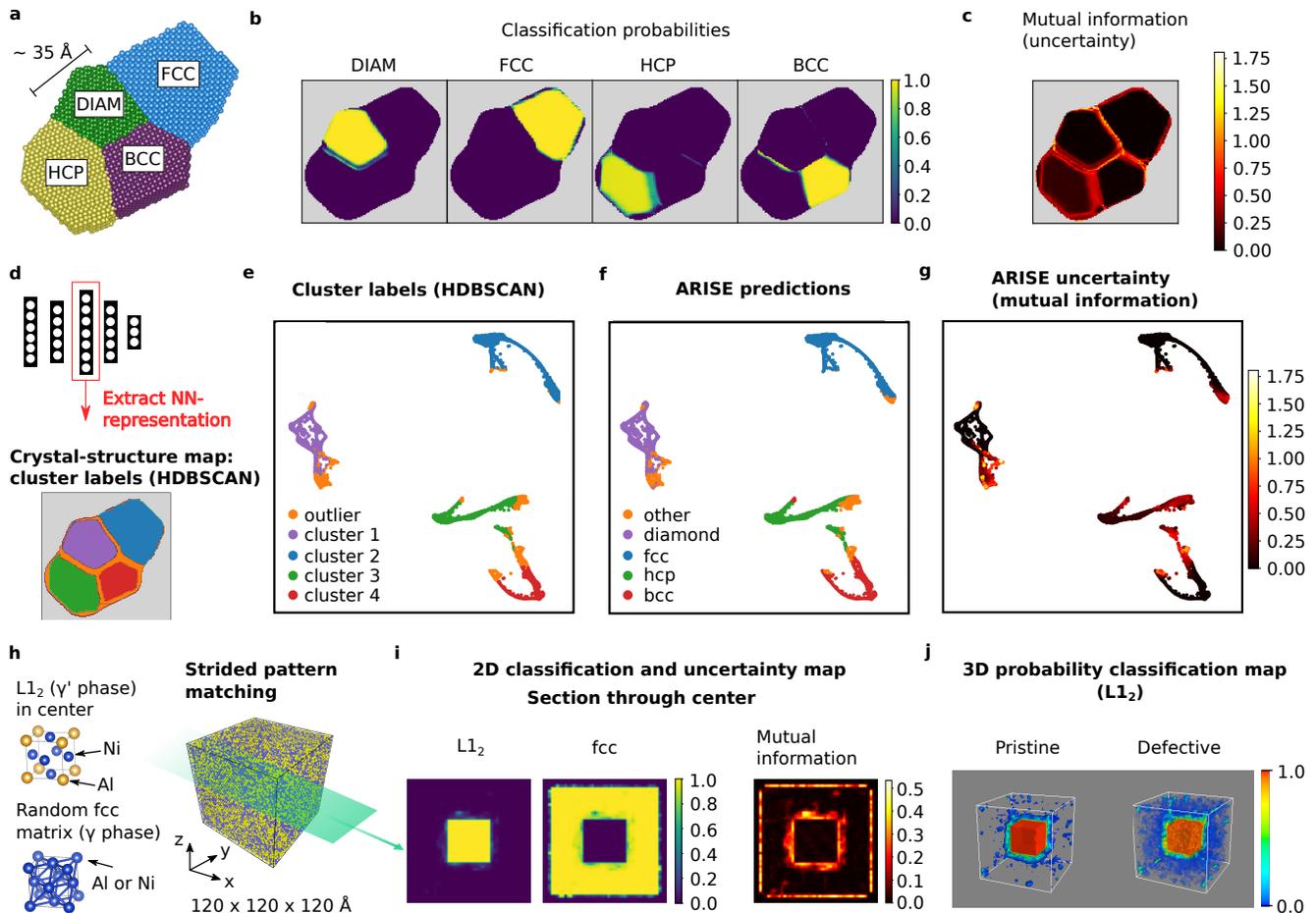


Figure 2. Analysis of synthetic polycrystals. **a** Mono-species polycrystal consisting of four grains with different symmetries. **b** Classification probabilities of expected prototypes. **c** Mutual information map for uncertainty quantification. **d-g** Unsupervised analysis of internal neural-network representations. **d** The neural-network representations are extracted for each local segment in **a** (obtained via SPM). Clustering (via HDBSCAN) is applied to this high-dimensional internal space; the polycrystal is marked according to the resulting clusters (see legend in **e** for the color assignments). **e-g** Two-dimensional projection (via UMAP) of neural-network representations colored by cluster label, ARISE predicted class, and mutual information, respectively. In **e**, all points for which HDBSCAN does not assign a cluster are labeled as “outlier”. In **f**, all points that are not classified as fcc, diamond, hcp or bcc are labeled as “other”. Note that while the distances between points are meaningful, the axes merely serve as a bounding window and are not interpretable – a situation typically encountered in non-linear methods such as UMAP (cf. section 6⁵³). **h-j** Precipitate detection in Ni-based superalloys. **h** Binary model system (right) and depiction of the two appearing phases (left). **i** Classification probabilities of expected prototypes and mutual information for a slice through the center of the structure. **j** 3D-resolved detection of the precipitate via the L_{12} classification probability for the pristine (left) and highly-defective case (right), for which 20% of the atoms are removed and randomly displaced (up to 5% of the nearest neighbor distance).

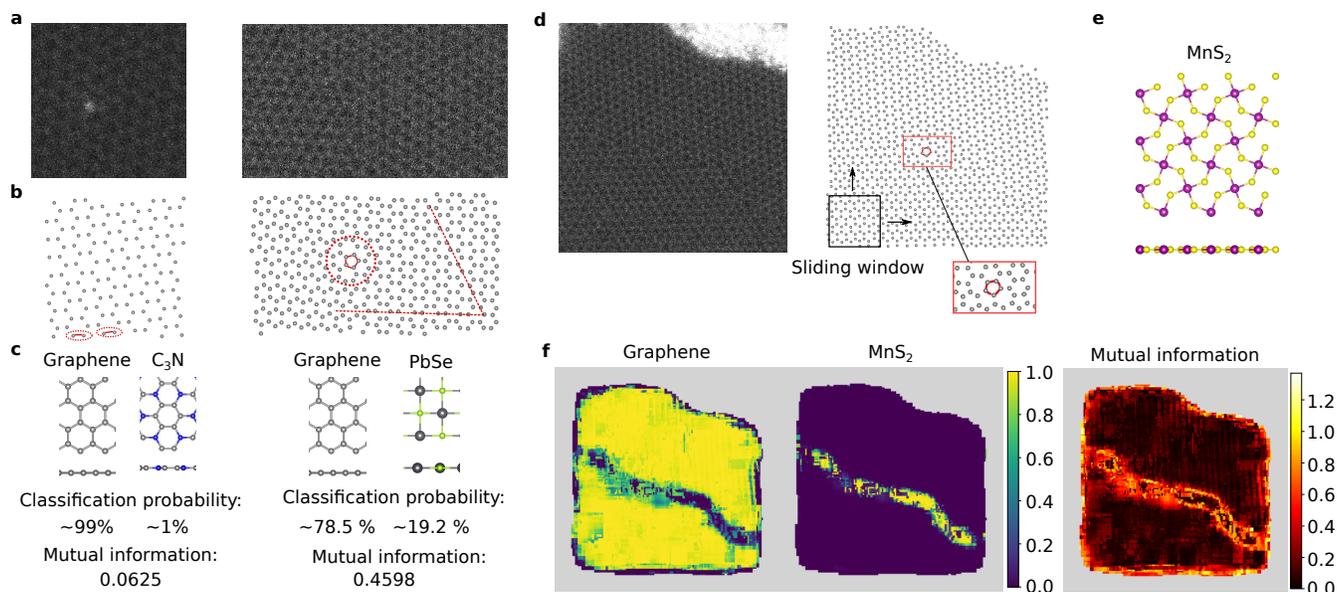


Figure 3. Analysis of HAADF images via ARISE and SPM. **a** Experimental HAADF images of two graphene structures. **b** The atomic positions are reconstructed from the images via AtomNet¹⁵. **c** The resulting atomic structures are analyzed using ARISE. The top predicted structures are shown. Mutual information is used to quantify the classification uncertainty. **d** HAADF image and reconstructed atomic positions (analogous to **a-b**) of a larger sample. Pentagons can be spotted near the grain boundary (see inset). **e** MnS₂ prototype. **f** Local analysis via strided pattern matching: graphene is the dominant structure. Different prototypes (MnS₂) are only assigned - and with high uncertainty (mutual information) - at the grain boundary.

tetragonal L1₀ phase^{59,60}. Besides the expected prototypes, ARISE also finds regions similar to tetragonally distorted, mono-species fcc (cf. Fig. S10), which is meaningful given the presence of fcc and the tetragonal phase L1₀.

To go beyond the information provided by classification and discover hidden patterns and trends in AET data, we conduct an exploratory analysis using unsupervised learning on ARISE's internal representations. While the procedure is similar to the one presented in Fig. 2d-g, here the analysis is truly exploratory (no ground truth is known), and data comes from experiment. First, all SPM boxes classified as L1₀ are extracted, this choice motivated by the physical relevance of this phase, in particular due to its magnetic properties⁵⁹. This reduces the number of data points (boxes) from 43679 to 5359 – a significant filtering step for which the automatic nature of ARISE is essential. In the representation space of the first hidden layer, HDBSCAN identifies six clusters (and the outliers). To interpret the cluster assignments, we analyze geometrical characteristics of atomic structures (i.e., the local boxes) assigned to the different clusters. Specifically, we consider the nearest neighbor distances between Fe and Pt atoms, d_{FeFe} and d_{PtPt} , respectively (cf. section S1.1 for the definition). For an ideal tetragonal structure, the difference $\Delta d = d_{\text{FeFe}} - d_{\text{PtPt}}$ is zero (cf. Fig. 4 b, top left); a deviation from this value thus quantifies the level of distortion. Looking at the histograms of the (signed) quantity Δd shown in Fig. 4b for each cluster, one can observe that each distribution is peaked; moreover, the distribution centers vary from negative

to positive Δd values across different clusters. The distribution of the outliers is shown for comparison: the Δd distribution is very broad, since outlier points are not a meaningful cluster. While overlap exists, the clusters correspond to subgroups of structures, each distorted in a different way, as quantified by Δd . Thus, we discovered a clear trend via the cluster assignment that correlates with the level of distortion. The cluster separation can be visualized in 2D via UMAP (cf. Fig. 4b). Notably, the clusters do not overlap, even in this highly compressed representation (from 256 to 2 dimensions). Some of the clusters may also contain further sub-distributions, which seems apparent for instance from the Δd distribution of cluster 6. The regions corresponding to the clusters could be hinting at a specific growth mechanism of the L1₀ phase during annealing, although further investigations are necessary to support this claim. The present analysis provides a protocol for the machine-learning driven exploration of structural data: supervised learning is employed to filter out a class of interest (which is not a necessary step, cf. Fig. 2d-g), then unsupervised learning is applied to the NN representations, revealing regions sharing physically meaningful geometrical characteristics.

Finally, we apply ARISE to time-resolved (4D) AET data. Specifically, a nanoparticle measured for three different annealing times is investigated⁶¹. The mutual information as obtained via SPM is shown in Fig. 4d for five central slices. In regions between outer shell and inner core, the mutual information clearly decreases for larger annealing times, indicating

that crystalline order increases inside the nanoparticle (see also Fig. S11 for more details). This analysis confirms that the predictive uncertainty of ARISE, as quantified by the mutual information, directly correlates with crystalline order. The mutual information can be therefore considered an AI-based order parameter, which we anticipate to be useful in future nucleation dynamics studies.

Discussion

In this work, Bayesian deep learning is employed to achieve a flexible, robust and threshold-independent crystal classification model, which we term ARISE. This approach correctly classifies a comprehensive and diverse set of crystal structures from computations and experiments, including polycrystalline systems (via strided pattern matching). ARISE is trained on ideal synthetic systems only and correctly identifies crystal structures in STEM and AET experiments, hence demonstrating strong generalization capabilities. The Bayesian deep-learning model provides classification probabilities, which - at variance with standard NNs - allow for the quantification of predictive uncertainty via mutual information. The mutual information is found to directly correlate with the degree of crystalline order, as shown by the analysis of time-resolved data from AET experiments. The internal NN representations are analyzed via state-of-the-art unsupervised learning. The clusters identified in this high-dimensional internal space allow to uncover physically meaningful structural regions. These can be grain boundaries, but also unexpected substructures sharing geometrical properties, as shown for metallic nanoparticles from AET experiments. This illustrates how supervised and unsupervised machine learning can be combined to discover hidden patterns in materials science data. In particular, the physical content learned by the NN model is explained by means of unsupervised learning. Since ARISE is not limited to predicting the space group, systems where the space group does not characterize the crystal structure can be tackled (as demonstrated for carbon nanotubes). More complex systems such as quasi-crystals⁶², periodic knots, or weavings⁶³ could also be considered. Indeed, ARISE can be applied to any data providing Cartesian coordinates labeled by chemical species. Practically, one simply needs to add the new structures of interest to the training set, and re-train or fine-tune (i.e., via transfer learning) the NN with the desired labels. Moreover, the mutual information allows to quantify the defectiveness of a structure; this could be exploited to automatically evaluate the quality of STEM images, for example one may automatically screen for STEM images that are likely to contain structural defects. Applications in active learning⁶⁴ for materials science are also envisioned, where uncertainty is crucial for example when deciding on the inclusion of additional - typically computationally costly - points in the dataset.

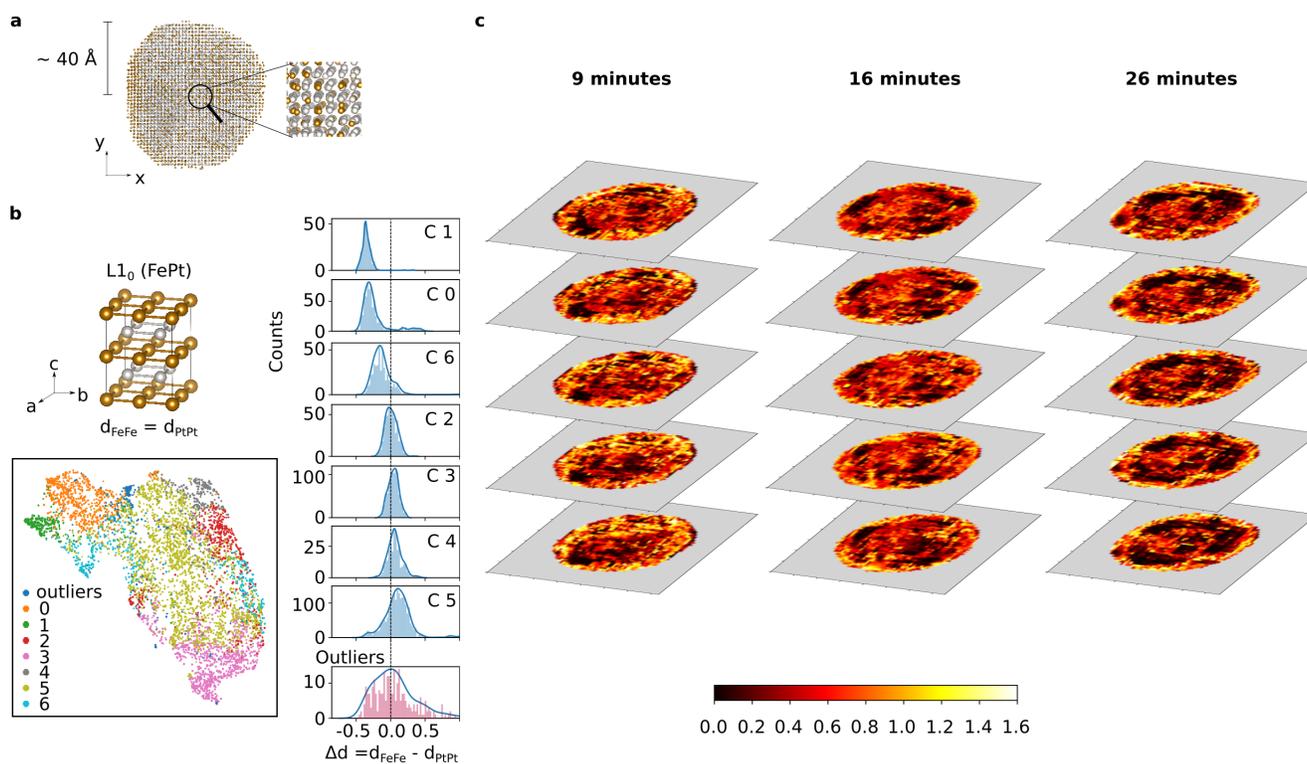


Figure 4. Analysis of atomic electron tomography data. . **a** Side view of FePt nanoparticle ($\sim 23\text{k}$ atoms), with atomic positions and chemical species from AET data⁶⁰. **b** Two-dimensional projection (bottom left) of neural-network representations (first hidden layer) via UMAP for regions classified as $L1_0$ by ARISE. The distribution of the difference between the nearest neighbor distances d_{FeFe} and d_{PtPt} (highlighted by bonds in top left part) is shown for each cluster (right). **c** Five central slices (mutual information, obtained via SPM) for three different annealing times (data from 4D AET experiment⁶¹).

Methods

Dataset creation. To compute the training set (39204 data points in total), we include periodic and non-periodic systems. For the former, no supercells are necessary (as SOAP is supercell-invariant for periodic structures). For the latter, a given structure (or rather its unit cell as obtained from the respective database) is isotropically replicated until at least 100 atoms are contained in the structure. Then this supercell structure and the next two larger isotropic replicas are included. With this choice of system sizes, we focus on slab- and bulk-like systems. Note that the network may not generalize to non-periodic structures outside the chosen supercell range. Practically, if the need to classify much smaller or larger supercells arises, one can include additional replicas to the training set and retrain the model (while for larger supercells it is expected that the network will generalize, see also Fig. S5). Retraining is computationally easy due to fast convergence time. Note that for 2D structures, only in-plane replicas are considered.

Elemental solids and binary compounds are selected from the AFLOW library of crystallographic prototypes⁸. Ternary, quaternary, and 2D materials are taken from the computational materials repository (CMR)⁴⁹.

Nanotubes are created using the atomic simulation environment (ASE)⁶⁵ where the chiral numbers (n,m) provide the class labels. We filter out chiral indices (n, m) (with the integer values n,m taking values in [0, 10]) for which the diameter is in the range [4 Å, 6 Å] (and skip the cases where $n = m = 0$, $n < m$). Then, we increase the length of each nanotube until at least 100 atoms are contained. No additional lengths are included as it was checked that there is no major change in the SOAP descriptor (via calculating the cosine similarity between descriptors representing nanotubes of different length). For more complex nanotubes (for instance, multi-walled systems), this may change.

For the cutoff R_C , we select the range $[3.0 \cdot d_{NN}, 5.0 \cdot d_{NN}]$ in steps of $0.2 \cdot d_{NN}$ and for σ the values $[0.08 \cdot d_{NN}, 0.1 \cdot d_{NN}, 0.12 \cdot d_{NN}]$. We calculate the SOAP descriptor using the *quippy* package (<https://libatoms.github.io/QUIP>), where we choose $n_{\max} = 9$ and $l_{\max} = 6$ as limits for the basis set expansion, resulting in an averaged SOAP vector of length 316. Furthermore, we increase the dataset by varying the *extrinsic scaling factor*: For a given prototype, the value of d_{NN} will deviate from the pristine value in presence of defects. Thus, to inform the network that the computation of d_{NN} is erroneous, we scale each pristine prototype not only by $1.0 \cdot d_{NN}$ but also $0.95 \cdot d_{NN}$ and $1.05 \cdot d_{NN}$. We term the factors 0.95, 1.0, 1.05 *extrinsic scaling factors*. One may also see this procedure as a way to increase the training set.

To create defective structures, we explained in the main text (cf. Table 1) how defects (displacements, missing atoms) are introduced. Note that we use the term “missing atoms” and not “vacancies” since the percentages of removed atoms we consider are well beyond regimes found in real materials. Also note that displacements as high as 4% of the nearest

neighbor distance might already cause a transition to the liquid phase in some solids. Still, as noted in the Introduction, experimental and computational data often present levels of distortions which are comparable or even substantially exceed these regimes. We introduce defects for all pristine prototypes included in the training set (specifically, for the supercells that are determined as described in the first paragraph – for both periodic and non-periodic boundary conditions, while for nanotubes only non-periodic structures are used). Since the defects are introduced randomly, we run 10 iterations of defect creation on each prototype. Then we calculate SOAP for all of these defective structures for one specific parameter setting ($R_C = 4.0 \cdot d_{NN}$, $\sigma = 0.1 \cdot d_{NN}$, *extrinsic scaling factor* = 1.0), which corresponds to the center of the respective parameter ranges included in the training set. Finally, we obtain 5880 defective structures for each defect ratio. In total, we compute defectives structures for three defect types (missing atoms and displacements introduced both separately and combined) for eight different defect ratios, giving in total 141,120 defective data points.

Neural network architecture and training procedure.

At prediction time, we need to fix T , the number of forward-passes being averaged (cf. section S1.3). We chose $T = 10^3$ for all results except Fig. 3 and S2, for which we increase T to 10^5 in order to get stable assignments in case of high uncertainty and very low probability candidates (i.e., $< 1.0\%$). Still, the most similar prototypes can be obtained already with 10^3 iterations.

Training is performed using Adam optimization⁶⁶. The multilayer perceptron is implemented in Keras⁶⁷ using Tensorflow⁶⁸ as backend. Furthermore we optimize hyperparameters such as the number of layers using Bayesian optimization, specifically the Tree-structured Parzen estimator (TPE) algorithm as provided by the python library hyperopt⁵⁰ (cf. section S1.4 for more details).

The initial training set is split (80/20% training / validation split of pristine structures, performed using scikit-learn and random state of 42) and the accuracy on the validation set is used as the performance metric to be minimized via hyperopt (for 50 iterations). Fast convergence (followed by oscillations around high accuracy values) or divergence is typically observed, which is why we train for a fixed number of epochs (300) and save only the model with the best performance on the validation set. Training is performed on 1 GPU (Tesla Volta V100 32GB) on the Talos machine-learning cluster in collaboration with the Max Planck Computing and Data facility (MPCDF). We observe that accuracies around 99% can be reached after few iterations, with individual training runs converging within 20 minutes, depending on model complexity.

Practically, strong models are obtained via this procedure, while further fine-tuning can be made to reach perfect accuracies. First, we restrict to one setting of training parameters (see previous section). From a computational efficiency point of view, this is also the preferred choice since one has

to compute only one descriptor per structure during prediction time. We select $R_C = 4.0 \cdot d_{NN}$ and $\sigma = 0.1 \cdot d_{NN}$ as well as an extrinsic scaling factor of 1.0. These choices are at the center of the respective parameter ranges. While the model with highest validation accuracy (on the whole training set) determined via hyperopt usually gives very strong performance, it is not necessarily the best possible one, especially in terms of generalization ability to defective structures. To find the optimal (i.e., most robust) model we select some of the best models (e.g. top 15) found via hyperopt and rank them based on their performance on pristine and defective structures (again for one setting of R_C, σ). In particular, we restrict to defective points with either $\leq 5\%$ atoms missing or $< 1\%$ atomic displacement, which comprises 35280 data points (six different defect ratios with 5880 points each). The number of pristine data points is 396. Using this strategy, we can identify a model with 100% accuracy on pristine and defective structures, which is reported in the last line of Table 1. The accuracy on the whole training set comprising 39204 data points is 99.66%.

We also investigate the performance on higher defect ratios beyond physically reasonable perturbations, since this is typically encountered in atom-probe experiments. In particular, we investigate three defect types (missing atoms, displacements, and both of them) comprising 105840 data points. The results for missing atoms ($> 5\%$) and displacements ($> 0.6\%$) can be found in Tables 1 and S2. Classification accuracies on structures with both missing atoms and displacements are specified in Table S3. Note that training and model selection only on pristine structures can yield robust models, especially if the number of classes is reduced. For instance, training only train on binary systems using a pristine set of 4356 data points (full SOAP parameter range) gives perfect accuracy on both the full training set and 3960 defective structures (displacements $\leq 0.06\%$ and $\leq 5\%$ missing atoms – for the setting $R_C = 4.0 \cdot d_{NN}$, $\sigma = 0.1 \cdot d_{NN}$, extrinsic scaling factor 1.0). Note that in general, if fewer classes are considered (e.g. ~ 20), the training time can be significantly reduced (e.g., to a few minutes).

Unsupervised learning: clustering and dimensionality reduction. HDBSCAN^{51,52} is a density-based, hierarchical clustering algorithm (see also the online documentation <https://hdbscan.readthedocs.io/en/latest/>). The final (so-called flat) clustering is derived from a hierarchy of clusters. The most influential parameter is the minimum cluster size that determines the minimum number of data points a cluster has to contain – otherwise it will be considered an outlier, i.e., not being part of any cluster. Practically, one can test a range of values for the minimum cluster size, in particular very small, intermediate and large ones – for instance for the results on the synthetic polycrystal in 2a, we test the values $\{25, 50, 100, 250, 500, 1000\}$. In line with intuition, the number of clusters grows (shrinks) for smaller (larger) values of minimum cluster size. A coherent picture with 4 clusters and clear boundaries (as indicated by

the outliers) arises for minimum cluster size values of around 500, for which we report the results in Fig. 2d-g and Fig. S6. For the nanoparticle data discussed in Fig. 4c, we observe that most of the points are considered outliers since the data contains substantially more distortions. To address this, we use the soft clustering feature of HDBSCAN, which allows to calculate a vector for each data point whose i -th component quantifies the probability that the given data point is member of cluster i . Then, we can infer a cluster assignment for points that would normally be considered outliers, by selecting for each point the cluster whose membership probability is maximal (while considering a point an outlier if all probabilities are below a certain threshold for which we choose 10%). For the minimum cluster size, we find that for values below 10 the number of clusters quickly grows while shrinking for larger values. We report the results for a minimum cluster size of 10 in Fig. 4c.

To visualize the clustering results, we use the manifold-learning technique UMAP⁵³ (see also the online documentation <https://umap-learn.readthedocs.io/en/latest/>). This method uses techniques from Riemannian geometry and algebraic topology to capture both the global and local structure of a manifold that underlies a given dataset. One of the most important parameters is the number of neighbors that will be considered to construct a topological representation of the data, where a small value takes only the local structure into account, while a large value considers the global relations between data points. We choose values of 500 for Fig. 2e-g and 50 for 4c, above which the 2D embeddings do not change significantly.

Synthetic polycrystal generation The structure in Fig. 2a is generated via the open-source software AtomsK⁶⁹.

Strided pattern matching parameters. Two parameters are most important for strided pattern matching analysis: Firstly, the stride defines the resolution and may be chosen arbitrarily small or large to increase or decrease the visualization of structural features. Note that the sliding allows us to discover smooth transitions, while the smoothness is determined by the step size. This way, boundary effects between neighbored local regions are reduced compared to the case of slightly or non-overlapping boxes (e.g., in the simple voxelization case). In particular, a small stride (e.g. 1 Å) mitigates boundary effects due to the discretization, which otherwise can influence the final classification and uncertainty maps. SPM is trivially parallel by construction, thus allowing the time-efficient characterization of large systems. Clearly, in a naive implementation, this procedure scales cubically with stride size. Practically, one may choose a large stride (in particular if the structure size would exceed computing capabilities) to obtain low-resolution classification maps, which may suffice to identify regions of interest. Then, one may zoom into these areas and increase the stride to obtain high resolution classification maps revealing more intricate features. Secondly, the box size determines the locality, i.e., the amount of structure that is “averaged” to infer the crystallo-

graphic prototype being most similar to a given local region. If this parameter is chosen too large, possibly interesting local features may be smoothed out. We recommend to use box sizes larger than 10-12Å, as in these cases, the number of contained atoms is typically within the range of the supercells the network is trained on (i.e., at least 100 atoms). The generalization ability to smaller structures depends on the prototype (cf. Fig. S5), and in general, if a smaller box size is desired while using our model, the practical solution is to add smaller supercells in the training set and retrain the network. Note that the shape of the local regions may be chosen to be different from boxes, e.g., spheres or any other shape that fits the application at hand. Moreover, we chose the grid in which the structure is strided to be cubic, while other discretizations are possible. Note that a one-dimensional striding can be applied to rod-like systems such as carbon nanotubes.

In this work, we choose the following SPM parameters: For the slab analysis in Fig. 2a, we choose a 1Å stride and a box size equal to the slab thickness (16Å). For the superalloy model system we choose the same box size but reduce the stride to 3Å, since this system is much larger and we want to demonstrate that for these systems, smaller strides still yield reasonable results. For the 2D STEM analysis (cf. Fig. 3f), we choose a stride of 4 (in units of pixels since atoms are reconstructed from images, while for typical graphene bond lengths of 1.42Å the relation $1\text{Å} \approx 8.5$ can be inferred). Moreover, we select a box size of 100 pixels ($\approx 12\text{Å}$). For the nanoparticle analysis, we choose a stride of 1Å and box size of 12Å for all of Fig. 4, except the clustering analysis, for which we reduce the stride to 2Å, to avoid an overcrowded 2D map. The box size of 16Å which allowed to distinguish chemically disordered fcc from ordered L1₂, cf. Fig. 2h-j) yields comparable results (see Fig. S10), while finding less L1₀ symmetry and more fcc since a larger amount of structure is averaged. Due to L1₀ showing special magnetic properties, we are interested in having a larger pool of candidate regions, which is why we choose a box size of 12Å (corresponding to the smallest value such that the average number of atoms in each box is greater than 100).

Atomic electron tomography.

ARISE's predictions are reliable since all the symmetries that typically occur in FePt nanoparticles are included in the training set – except the disordered phase for which it has been demonstrated in the analysis of the Ni-based superalloy model system that ARISE is sensitive to chemical ordering. Moreover, a supplementing study reveals that ARISE can analyze structural transformations, in particular similar to the ones taking place in nanoparticles (cf. section S2.2, where the so-called Bain path is investigated).

Due to diffusion, the shape of the three nanoparticles (cf. Fig. 4d-e) and thus the number of atoms is changing. Rough alignment of the nanoparticles was checked using point set registration: Specifically, we employed the coherent point drift algorithm⁷⁰ as implemented in the python package pycpd (<https://github.com/siavashk/pycpd>). We

extracted only the core of the nanoparticle, which is reported to remain similar during the annealing procedure¹⁷. After applying the algorithm, the remaining mismatch is negligible (3-10° for all three Euler angles).

Data availability

The training and test dataset, and additional relevant data can be found at <https://github.com/angeloziletti/ai4materials>. The experimental HAADF images in this work are taken from the notebooks in the repository <https://github.com/pycroscopy/AICrystallographer/tree/master/AtomNet>. AET data is taken from the original references and <https://www.materialsdatbank.org/>.

Code availability

A Python code library “ai4materials” is available at <https://github.com/angeloziletti/ai4materials>. In more detail, ai4materials provides tools to perform complex analysis of materials science data using machine learning techniques. Furthermore, functions for pre-processing, saving, and loading of materials science data are provided, with the goal to ease traceability, reproducibility, and prototyping of new models.

An online tutorial to reproduce the main results presented in this work can be found in the NOMAD Analytics-Toolkit (<https://analytics-toolkit.nomad-coe.eu/tutorial-ARISE>).

References

1. Baddoo, N. Stainless steel in construction: A review of research, applications, challenges and opportunities. *J. Constr. Steel Res.* **64**, 1199–1206 (2008).
2. Herbig, M. *et al.* Atomic-scale quantification of grain boundary segregation in nanocrystalline material. *Phys. Rev. Lett.* **112**, 126103 (2014).
3. Meiners, T., Frolov, T., Rudd, R. E., Dehm, G. & Lieb-scher, C. H. Observations of grain-boundary phase transformations in an elemental metal. *Nat.* **579**, 375–378 (2020).
4. Ferrari, A. C. *et al.* Science and technology roadmap for graphene, related two-dimensional crystals, and hybrid systems. *Nanoscale* **7**, 4598–4810 (2015).
5. De Volder, M. F., Tawfick, S. H., Baughman, R. H. & Hart, A. J. Carbon nanotubes: present and future commercial applications. *Sci.* **339**, 535–539 (2013).
6. Draxl, C. & Scheffler, M. NOMAD: The FAIR concept for big data-driven materials science. *MRS Bull.* **43**, 676–682 (2018).
7. Draxl, C. & Scheffler, M. The NOMAD laboratory: from data sharing to artificial intelligence. *J. Physics: Mater.* **2**, 036001 (2019).

Layer type	Specifications
Input Layer + Dropout	Materials representation (SOAP descriptor, 316 components)
Dense Layer + Dropout + ReLU	Size: 256
Dense Layer + Dropout + ReLU	Size: 512
Dense Layer + Dropout + ReLU	Size: 256
Dense Layer + Softmax	Size: 108 (= # classes)

Table 2. Architecture of the fully-connected Bayesian neural network used in this work. The dropout ratio is 3.17% for all layers. The total number of parameters is 371,820. While training time was fixed to 300 epochs, hyperopt found a batch size of 64 and a learning rate of $2.16 \cdot 10^{-4}$.

8. Mehl, M. J. *et al.* The AFLOW library of crystallographic prototypes: part 1. *Comput. Mater. Sci.* **136**, S1–S828 (2017).
9. Hicks, D. *et al.* The AFLOW Library of Crystallographic Prototypes: part 2. *Comput. Mater. Sci.* **161**, S1–S1011 (2019).
10. Saal, J. E., Kirklin, S., Aykol, M., Meredig, B. & Wolverton, C. Materials design and discovery with high-throughput density functional theory: the open quantum materials database (OQMD). *JOM* **65**, 1501–1509 (2013).
11. Jain, A. *et al.* A high-throughput infrastructure for density-functional theory calculations. *Comput. Mater. Sci.* **50**, 2295–2310 (2011).
12. Haastrup, S. *et al.* The Computational 2D Materials Database: high-throughput modeling and discovery of atomically thin crystals. *2D Mater.* **5**, 042002 (2018).
13. Mounet, N. *et al.* Two-dimensional materials from high-throughput computational exfoliation of experimentally known compounds. *Nat. Nanotechnol.* **13**, 246 (2018).
14. Pennycook, S. J. & Nellist, P. D. *Scanning Transmission Electron Microscopy: Imaging and Analysis* (Springer New York, 2011).
15. Ziatdinov, M. *et al.* Deep learning of atomically resolved scanning transmission electron microscopy images: chemical identification and tracking local transformations. *ACS nano* **11**, 12742–12752 (2017).
16. Gault, B., Moody, M. P., Cairney, J. M. & Ringer, S. P. Atom probe crystallography. *Mater. Today* **15**, 378–386 (2012).
17. Zhou, J., Yang, Y., Ercius, P. & Miao, J. Atomic electron tomography in three and four dimensions. *MRS Bull.* **45**, 290–297 (2020).
18. Gault, B. A brief overview of atom probe tomography research. *Appl. Microsc.* **46**, 117–126 (2016).
19. Togo, A. & Tanaka, I. Spglib: a software library for crystal symmetry search. *arXiv preprint arXiv:1808.01590* (2018).
20. Hicks, D. *et al.* AFLOW-SYM: platform for the complete, automatic and self-consistent symmetry analysis of crystals. *Acta Crystallogr. Sect. A: Foundations Adv.* **74**, 184–203 (2018).
21. Honeycutt, J. D. & Andersen, H. C. Molecular dynamics study of melting and freezing of small Lennard-Jones clusters. *J. Phys. Chem.* **91**, 4950–4963 (1987).
22. Stukowski, A. Structure identification methods for atomistic simulations of crystalline materials. *Model. Simul. Mater. Sci. Eng.* **20**, 045021 (2012).
23. Ackland, G. & Jones, A. Applications of local crystal structure measures in experiment and simulation. *Phys. Rev. B* **73**, 054104 (2006).
24. Larsen, P. M., Schmidt, S. & Schiøtz, J. Robust structural identification via polyhedral template matching. *Model. Simul. Mater. Sci. Eng.* **24**, 055007 (2016).
25. Goodfellow, I., Bengio, Y. & Courville, A. *Deep Learning* (MIT Press, 2016). <http://www.deeplearningbook.org>.
26. Geiger, P. & Dellago, C. Neural networks for local structure detection in polymorphic systems. *The J. Chem. Phys.* **139**, 164105 (2013).
27. Reinhart, W. F., Long, A. W., Howard, M. P., Ferguson, A. L. & Panagiotopoulos, A. Z. Machine learning for autonomous crystal structure identification. *Soft Matter* **13**, 4733–4745 (2017).
28. Dietz, C., Kretz, T. & Thoma, M. Machine-learning approach for local classification of crystalline structures in multiphase systems. *Phys. Rev. E* **96**, 011301 (2017).
29. Ziletti, A., Kumar, D., Scheffler, M. & Ghiringhelli, L. M. Insightful classification of crystal structures using deep learning. *Nat. Commun.* **9**, 1–10 (2018).
30. Gal, Y. & Ghahramani, Z. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, 1050–1059 (2016).
31. Gal, Y. *Uncertainty in deep learning*. Ph.D. thesis, University of Cambridge (2016).
32. Behler, J. Atom-centered symmetry functions for constructing high-dimensional neural network potentials. *The J. Chem. Phys.* **134**, 074106 (2011).

33. Bartók, A. P., Payne, M. C., Kondor, R. & Csányi, G. Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons. *Phys. Rev. Lett.* **104**, 136403 (2010).
34. Bartók, A. P., Kondor, R. & Csányi, G. On representing chemical environments. *Phys. Rev. B* **87**, 184115 (2013).
35. Huo, H. & Rupp, M. Unified representation for machine learning of molecules and crystals. *arXiv preprint arXiv:1704.06439* (2017).
36. Shapeev, A. V. Moment tensor potentials: A class of systematically improvable interatomic potentials. *Multiscale Model. & Simul.* **14**, 1153–1173 (2016).
37. Bartók, A. P. & Csányi, G. Gaussian approximation potentials: A brief tutorial introduction. *Int. J. Quantum Chem.* **115**, 1051–1057 (2015).
38. De, S., Bartók, A. P., Csányi, G. & Ceriotti, M. Comparing molecules and solids across structural and alchemical space. *Phys. Chem. Chem. Phys.* **18**, 13754–13769 (2016).
39. Rosenbrock, C. W., Homer, E. R., Csányi, G. & Hart, G. L. Discovering the building blocks of atomic systems using machine learning: application to grain boundaries. *npj Comput. Mater.* **3**, 1–7 (2017).
40. Bengio, Y., Courville, A. & Vincent, P. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis Mach. Intell.* **35**, 1798–1828 (2013).
41. Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580* (2012).
42. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The J. Mach. Learn. Res.* **15**, 1929–1958 (2014).
43. Ashcroft, N. W. & Mermin, N. D. *Solid State Physics* (Cengage Learning, London, 2011).
44. Pettifor, D. G. *Bonding and structure of molecules and solids* (Oxford University Press, 1995).
45. Reed, R. C. *The Superalloys: Fundamentals and Applications* (Cambridge University Press, 2008).
46. Castelli, I. E., Thygesen, K. S. & Jacobsen, K. W. Calculated optical absorption of different perovskite phases. *J. Mater. Chem. A* **3**, 12343–12349 (2015).
47. Pandey, M. & Jacobsen, K. W. Promising quaternary chalcogenides as high-band-gap semiconductors for tandem photoelectrochemical water splitting devices: A computational screening approach. *Phys. Rev. Mater.* **2**, 105402 (2018).
48. Novoselov, K., Mishchenko, A., Carvalho, A. & Neto, A. C. 2D materials and van der Waals heterostructures. *Sci.* **353**, aac9439–aac9439 (2016).
49. Landis, D. D. *et al.* The computational materials repository. *Comput. Sci. & Eng.* **14**, 51–57 (2012).
50. Bergstra, J., Yamins, D. & Cox, D. D. Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28, ICML'13*, I–115–I–123 (JMLR.org, 2013).
51. McInnes, L. & Healy, J. Accelerated hierarchical density based clustering. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, 33–42 (IEEE, 2017).
52. McInnes, L., Healy, J. & Astels, S. HDBSCAN: Hierarchical density based clustering. *The J. Open Source Softw.* **2**, 205 (2017). DOI 10.21105/joss.00205.
53. McInnes, L., Healy, J. & Melville, J. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426* (2018).
54. Parsa, A. B. *et al.* Advanced Scale Bridging Microstructure Analysis of Single Crystal Ni-Base Superalloys. *Adv. Eng. Mater.* **17**, 216–230 (2015). DOI 10.1002/adem.201400136.
55. Miao, J., Ercius, P. & Billinge, S. J. Atomic electron tomography: 3D structures without crystals. *Sci.* **353**, aaf2157–aaf2157 (2016).
56. Chen, C.-C. *et al.* Three-dimensional imaging of dislocations in a nanoparticle at atomic resolution. *Nat.* **496**, 74–77 (2013).
57. Xu, R. *et al.* Three-dimensional coordinates of individual atoms in materials revealed by electron tomography. *Nat. Mater.* **14**, 1099–1103 (2015).
58. Tian, X. *et al.* Correlating the three-dimensional atomic defects and electronic properties of two-dimensional transition metal dichalcogenides. *Nat. Mater.* 1–7 (2020).
59. Sun, S. Recent advances in chemical synthesis, self-assembly, and applications of FePt nanoparticles. *Adv. Mater.* **18**, 393–403 (2006).
60. Yang, Y. *et al.* Deciphering chemical order/disorder and material properties at the single-atom level. *Nat.* **542**, 75–79 (2017).
61. Zhou, J. *et al.* Observing crystal nucleation in four dimensions using atomic electron tomography. *Nat.* **570**, 500–503 (2019).
62. Levine, D. & Steinhardt, P. J. Quasicrystals: a new class of ordered structures. *Phys. Rev. Lett.* **53**, 2477 (1984).
63. Liu, Y., O’Keeffe, M., Treacy, M. M. & Yaghi, O. M. The geometry of periodic knots, polycatenanes and weav-

- ing from a chemical perspective: a library for reticular chemistry. *Chem. Soc. Rev.* **47**, 4642–4664 (2018).
64. Gal, Y., Islam, R. & Ghahramani, Z. Deep Bayesian active learning with image data. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 1183–1192 (JMLR. org, 2017).
 65. Larsen, A. H. *et al.* The atomic simulation environment—a Python library for working with atoms. *J. Physics: Condens. Matter* **29**, 273002 (2017).
 66. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
 67. Chollet, F. *et al.* Keras. <https://keras.io> (2015).
 68. Abadi, M. *et al.* TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems (2015). URL <https://www.tensorflow.org/>.
 69. Hirel, P. AtomsK: a tool for manipulating and converting atomic data files. *Comput. Phys. Commun.* **197**, 212–219 (2015).
 70. Myronenko, A. & Song, X. Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis Mach. Intell.* **32**, 2262–2275 (2010).
 71. Mavračić, J., Mocanu, F. C., Deringer, V. L., Csányi, G. & Elliott, S. R. Similarity between amorphous and crystalline phases: The case of TiO₂. *The J. Phys. Chem. Lett.* **9**, 2985–2990 (2018).
 72. Houlisby, N., Huszár, F., Ghahramani, Z. & Lengyel, M. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745* (2011).
 73. Bergstra, J. S., Bardenet, R., Bengio, Y. & Kégl, B. Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems*, 2546–2554 (2011).
 74. Stukowski, A. Visualization and analysis of atomistic simulation data with OVITO—the Open Visualization Tool. *Model. Simul. Mater. Sci. Eng.* **18**, 015012 (2009).
 75. Bain, E. C. & Dunkirk, N. Y. The nature of martensite. *trans. AIME* **70**, 25–47 (1924).
 76. Zhao, J.-C. & Notis, M. R. Continuous cooling transformation kinetics versus isothermal transformation kinetics of steels: a phenomenological rationalization of experimental observations. *Mater. Sci. Eng. R: Reports* **15**, 135–207 (1995).
 77. Grimvall, G., Magyari-Köpe, B., Ozoliņš, V. & Persson, K. A. Lattice instabilities in metallic elements. *Rev. Mod. Phys.* **84**, 945 (2012).
 78. Alippi, P., Marcus, P. M. & Scheffler, M. Strained tetragonal states and bain paths in metals. *Phys. Rev. Lett.* **78**, 3892–3895 (1997). DOI 10.1103/PhysRevLett.78.3892.
 79. Buschbeck, J. *et al.* Full Tunability of Strain along the fcc-bcc Bain Path in Epitaxial Films and Consequences for Magnetic Properties. *Phys. Rev. Lett.* **103**, 216101 (2009). DOI 10.1103/PhysRevLett.103.216101.
 80. Haftel, M. I. & Gall, K. Density functional theory investigation of surface-stress-induced phase transformations in fcc metal nanowires. *Phys. Rev. B - Condens. Matter Mater. Phys.* **74**, 035420 (2006). DOI 10.1103/PhysRevB.74.035420.
 81. Davey, W. P. The lattice parameter and density of pure tungsten. *Phys. Rev.* **26**, 736 (1925).
 82. Straumanis, M. & Yu, L. Lattice parameters, densities, expansion coefficients and perfection of structure of Cu and of Cu-In α phase. *Acta Crystallogr. Sect. A: Cryst. Physics, Diffraction, Theor. Gen. Crystallogr.* **25**, 676–682 (1969).
 83. Deshpande, V. & Pawar, R. Anisotropic thermal expansion of indium. *Acta Crystallogr. Sect. A: Cryst. Physics, Diffraction, Theor. Gen. Crystallogr.* **25**, 415–416 (1969).
 84. Zachariasen, W. On the crystal structure of protactinium metal. *Acta Crystallogr.* **12**, 698–700 (1959).
 85. Isayev, O. *et al.* Materials cartography: representing and mining materials space using structural and electronic fingerprints. *Chem. Mater.* **27**, 735–743 (2015).

Acknowledgements

We acknowledge funding from BiGmax, the Max Planck Society’s Research Network on Big-Data-Driven Materials Science. L. M. G. acknowledges funding from the European Union’s Horizon 2020 research and innovation program, under grant agreements No. 676580 (NOMAD) and No. 740233 (TEC1p). Furthermore, the authors acknowledge the Max Planck Computing and Data facility (MPCDF) for computational resources and support, which enabled neural-network training on 1 GPU (Tesla Volta V100 32GB) on the Talos machine learning cluster. The authors thank Matthias Scheffler for initiating this research direction and providing comments to the manuscript.

Author contributions

All authors designed the project. A.Z. and A.L. wrote the code, A.L. performed all the calculations. A.Z. and L.M.G. supervised the project. All authors wrote and reviewed the manuscript.

Contents

References	13
1 Supplementary material	18
1.1 Isotropic scaling	18
1.2 SOAP descriptor	18
1.3 Bayesian deep learning	19
1.4 Hyperparameter optimization	19
2 Supplementary notes	19
2.1 Benchmarking	19
2.2 The Bain transformation path	21
2.3 When the model is forced to fail: analysis of ARISE out-of-sample predictions	21

1 Supplementary material

1.1 Isotropic scaling

To reduce the dependency on lattice parameters, we isotropically scale each prototype according to its nearest neighbor distance d_{NN} . This way, one degree of freedom is eliminated, implying that all cubic systems are equivalent and thus are correctly classified by construction. To compute d_{NN} , we calculate the radial distribution function, for which we determine in a first step the histogram of all neighbor distances. Since the area of spherical shells grows with the squared radius, we divide the histogram by the squared radial distance. Then, we use the center of the maximally populated bin as the nearest neighbor distance d_{NN} . Dividing the atomic position by d_{NN} yields the final isotropically scaled structure, which is used for calculating the SOAP descriptor. Alternatively, one may use the mean of the nearest neighbors as d_{NN} , which, however, is more prone to defects. In case of multiple chemical species, we consider all possible substructures as formed by the constituting species to calculate the SOAP descriptor (see section SOAP descriptor). For each of the substructures, we compute d_{NN} , while we determine the histogram of neighbor distances only from distances between atoms whose chemical species coincide with those of the substructure. For instance, given the substructure (α, β) , i.e., the atomic arrangement of atoms with species β as seen from the perspective of atoms with species α , we consider only α -atoms and determine all distances to β -atoms.

1.2 SOAP descriptor

As discussed in the main text, encoding of physical requirements we know to be true is crucial for machine-learning application. For instance, in crystal classification, two atomic structures that differ only by a rotation must have the same classification label. This is not guaranteed if real space atomic coordinates are used as descriptor (cf. Fig. 1a). As an attempt to fix this, one might include a discrete subset of orientations in the training set, hoping that the model will generalize to unseen rotations. However, there is no theoretical guarantee that the model will learn the rotational symmetry, and if it does not, it will fail to generalize and return different predictions for symmetrically equivalent structures. In contrast, when a rotationally invariant descriptor is employed, only one crystal orientation needs to be included in the training set and the model will generalize to all rotations by construction. This reasoning readily applies to other physics requirements such as translational, or permutation invariance (for atoms with same chemical species).

In the following, we provide details on adapting the standard SOAP descriptor such that its number of components is independent on the number of atoms and chemical species.

Starting with the simple case of one chemical species, we consider a local atomic environment \mathcal{X} , being defined by a cutoff region (with radius R_C) around a central atom, located at the origin of the reference frame. Each atom within this area is represented by a Gaussian function centered at the

atomic position \vec{r}_i and with width σ . Then, the local atomic density function of \mathcal{X} can be written as³⁴

$$\rho_{\mathcal{X}}(\vec{r}) = \sum_{i \in \mathcal{X}} \exp\left(-\frac{(\vec{r} - \vec{r}_i)^2}{2\sigma^2}\right) = \sum_{blm} c_{blm} u_b(r) Y_{lm}(\hat{r}), \quad (1)$$

where in the second step, an expansion in terms of spherical harmonics $Y_{lm}(\hat{r})$ and a set of radial basis functions $\{u_b(r)\}$ is performed. One can show that the rotationally invariant power spectrum is given by³⁴

$$p(\mathcal{X})_{b_1 b_2 l} = \pi \sqrt{\frac{8}{2l+1}} \sum_m (c_{b_1 l m})^\dagger c_{b_2 l m}. \quad (2)$$

These coefficients can be arranged in a normalized (SOAP) vector $\hat{p}(\mathcal{X})$, describing the local atomic environment \mathcal{X} . In total, we obtain as many SOAP vectors as atoms in the structure, which one can average to obtain a materials descriptor independent of the number atoms N_{at} . Another possibility (the standard setting in the software we use) is to average the coefficients c_{blm} first and then compute Eq. 2 from this⁷¹. The cutoff radius R_C and σ (cf. Eq. 1) are hyperparameters, i.e., supervised learning cannot be used directly to assign values to these parameters, while their specific choice will affect the results. Typically, one would employ cross-validation while here, we take a different route: First, we assess the similarity between SOAP descriptors using the cosine similarity to identify parameter ranges that provide sufficient contrast between the prototypes. Using this experimental approach, we find that values near $\sigma = 0.1 \cdot d_{\text{NN}}$ and $R_C = 4.0 \cdot d_{\text{NN}}$ yield good results. Then we augment our dataset with SOAP descriptors calculated for different parameter settings (see section Dataset creation).

The extension to several chemical species is achieved by considering all possible substructures as formed by the constituting atoms: Considering NaCl, we first inspect the lattice of Cl atoms as seen by the Na atoms, which we denote by (Na, Cl); this means that Na atoms are considered as central atoms in the construction of the local atomic environment while only Cl atoms are considered as neighbors. A similar construction is made for the remaining substructures (Na, Na), (Cl, Na), and (Cl, Cl), which may be quite similar depending on the atomic structure. For each substructure, we compute the SOAP vectors via Eq. 2, obtaining a collection of SOAP vectors. Averaging these gives us four (in case of NaCl) averaged SOAP vectors. Averaging the latter again, yields a materials representation being independent on the number of atoms and chemical species.

Formally, given a structure with S species $\alpha_1, \dots, \alpha_S$, we consider all substructures formed by pairs of species (α_i, α_j) , $j = 1, \dots, S$, resulting in S^2 averaged SOAP vectors $\langle \hat{p}_{\alpha_i \alpha_j} \rangle_{N_{\text{at}, \alpha_j}}$, where the bracket represents the average over number of atoms N_{at} of species α_i . These vectors are averaged over, yielding the final vectorial descriptor $\langle \langle \hat{p}_{\alpha_i \alpha_j} \rangle \rangle_{\alpha_i \alpha_j}$.

Note that this construction of SOAP deviates from the previously reported way of treating multiple chemical species in

the following way: Usually, for each atom, one constructs the following power spectra³⁸

$$p(\mathcal{X})_{b_1 b_2 l}^{\alpha\beta} = \pi \sqrt{\frac{8}{2l+1}} \sum_m (c_{b_1 l m}^\alpha)^\dagger c_{b_2 l m}^\beta, \quad (3)$$

where the coefficients originate from basis set expansion as in Eq. 2, while the density ρ is constructed separately for each species. For a specific α and β , the coefficients of Eq. 3 can be collected into vectors $\vec{p}_{\alpha\beta}$. In case of $\alpha \neq \beta$, cross-correlations, i.e., products of coefficients from different densities are used to construct the vectors $\vec{p}_{\alpha\beta}$, which are missing in our version.

1.3 Bayesian deep learning

As discussed in the main text, one can think of Bayesian neural networks as standard neural networks with distributions being placed over the model parameters. This results in probabilistic outputs from which principled uncertainty estimates can be obtained. The major drawback is that training and obtaining predictions from traditional Bayesian neural networks is generally difficult because it requires solving computationally costly high-dimensional integrals. For classification, expensive calculations are required to determine $p(y = c|x, D_{\text{train}})$, which is the probability that the classification is assigned to a class c , given input x and training data D_{train} . Then, for a specific input x (in our case the SOAP descriptor), the most likely class c , i.e., the one with largest $p(y = c|x, D_{\text{train}})$ is the predicted class.

Gal and Ghahramani³⁰ showed that stochastic regularization techniques such as dropout^{41,42} can be used to calculate high-quality uncertainty estimates (alongside predictions) at low cost. In dropout, neurons are randomly dropped in each layer before the network is evaluated for a given input. Usually, dropout is only used at training time with the goal of avoiding overfitting by preventing over-specialization of individual units. Keeping regularization also at test time allows to quantify the uncertainty. Practically, given a new input, one collects and subsequently aggregates the predictions while using dropout at prediction time. This gives a collection of probabilities being denoted as $p(y = c|x, \omega_t)$, which is the probability of predicting class c given the input x at a specific forward-pass t , with model parameters ω_t . From this collections of probabilities, one can estimate the actual quantity of interest, $p(y = c|x, D_{\text{train}})$, by a simple average³⁰:

$$p(y = c|x, D_{\text{train}}) \approx \frac{1}{T} \sum_{t=1}^T p(y = c|x, \omega_t), \quad (4)$$

where T is the number of forward-passes (see Methods section ‘‘Neural network architecture and training procedure’’ for details on how we choose this parameter). While the average can be used to infer the class label c , additional statistical information, which reflects the predictive uncertainty, is contained in the collected forward-passes, i.e., the probabilities $p(y = c|x, \omega_t)$ which effectively yield a histogram for each

class and define, when varying over all possible c , a (discrete) probability distribution. For instance, mutual information can be used to quantify the uncertainty from the expressions $p(y = c|x, \omega_t)$. Specifically, for a given test point x , the mutual information between the predictions and the model posterior $p(\omega|D_{\text{train}})$ (which captures the most probable parameters given the training data) is defined as^{30,72}

$$\begin{aligned} \mathbb{I}[y, \omega|x, D_{\text{train}}] \approx & \\ & - \sum_c \left(\frac{1}{T} \sum_t p(y = c|x, \omega_t) \right) \log \left(\frac{1}{T} \sum_t p(y = c|x, \omega_t) \right) \\ & + \frac{1}{T} \sum_c \sum_t p(y = c|x, \omega_t) \log p(y = c|x, \omega_t). \end{aligned} \quad (5)$$

1.4 Hyperparameter optimization

The Tree-structured Parzen estimator (TPE) algorithm^{50,73} is an example of a Bayesian optimization technique. Specifically, one has to define a search space which can comprise a variety of parameters such as the learning rate or model size specifics such as the number of layers or neurons. Then, the goal is to minimize a performance metric (in our case, we maximize the accuracy by minimizing its negative). For large search spaces, iterating through each possible combination, i.e., performing a grid search, will get expensive very quickly. Random search is one alternative, while Bayesian methods such as TPE can be more efficient. Approaches such as grid or random search assign uniform probability to each hyperparameter choice, which implies that a long time is spent at settings with low reward. This becomes particularly troublesome if the performance metric is expensive to calculate. In Bayesian methods such as TPE, the objective is replaced by a computationally cheaper surrogate model (for instance, Gaussian process or random forest regressor). New hyperparameters are selected iteratively in a Bayesian fashion. Specifically, the selection is based on an evaluation function (typically ‘‘expected improvement’’) taking into account the history of hyperparameter selections and thus avoiding corners of the search space with low reward.

The search space is chosen the following way (alongside the chosen hyperopt commands `hp.choice` or `hp.uniform`):

- Number of layers (2, 3, 4, 5), `hp.choice`
- Number of neurons in each layer (256 or 512), `hp.choice`
- Batch size, (64 or 128), `hp.choice`
- Learning rate, range (0.01,0.0001), `hp.uniform`
- Dropout rate, range (0.01, 0.05), `hp.uniform`

2 Supplementary notes

2.1 Benchmarking

In the following, we provide details on the benchmarking.

For `spglib`, we only include prototypes from AFLOW. The reason for excluding structures from the computational materials repository ‘‘CMR’’ is that we do not always have the

correct or meaningful labels for all structures. For instance, some 2D materials are specified as P1 in the database, which cannot be used as a correct label. Furthermore, for quaternary chalcogenides, the expected symmetries (as specified in the corresponding reference⁴⁷) cannot be reconstructed, which is most likely due to local optimization effects. Similar observations were made for the ternary Perovskites. More careful choice of precision parameters or additional local optimization may help. Thus, to enable a fair comparison, the benchmarking in the main text only reports results on elemental and binary compounds from AFLOW (where we know the true labels), while the performance on both AFLOW and CMR data is shown in table S1, S2, and S3. To avoid the impression that spglib is not applicable to ternary, quaternary, and 2D materials, we still provide the label “96/108” behind spglib methods in the benchmarking tables. Note that non-periodic structures are excluded for benchmarking, in particular carbon nanotubes, since these systems cannot be treated by spglib.

For the other benchmarking methods, which are common neighbor analysis (CNA, a-CNA), bond angle analysis (BAA), and polyhedral template matching (PTM), we use implementations provided in OVITO⁷⁴, where for BAA we apply the Ackland Jones method. As for spglib, only periodic structures were included. BAA, CNA, a-CNA all include fcc, bcc, and hcp structures, while PTM contains in addition sc, diamond, hexagonal diamond, graphene, graphitic boron nitride, L10, L12, zinc blende, and wurtzite. Each of the frameworks provide one label for each atom, i.e., for a structure with N atoms we obtain N labels. To obtain an accuracy score, we compare these N predictions to N true labels, which correspond to the space group associated with the prototype label (e.g., 194 for hcp). For CNA, we select the standard cutoff (depending on its value one is able to detect bcc but not hcp and vice versa). Also for BAA (Ackland Jones) and a-CNA standard settings are used. For PTM, an RMSD cutoff of 0.1 was used (again default in OVITO). Note that PTM can also distinguish different sites of the L12 structure. For simplicity, we did not label the L12 structure by sites and take this classification into account, but always assigning a true label as soon as an atom was assigned to the L12 class (even if it might be not the correct site).

Furthermore, for ARISE periodic and non-periodic structures are included, while for the benchmarking methods only periodic structures are considered. While for spglib, translational symmetry is violated by construction, the other methods can in principle be applied to these systems. However, when calculating the accuracy for a given non-periodic structure, we have to choose a label for the boundary atoms. If we select the same label for these atoms as for the central ones (which have a sufficiently larger number of neighbors), these methods will usually predict the class 'None' and interpreting this as a “misclassification” would decrease the total classification accuracy. Therefore, for a fair comparison, we exclude non-periodic structures.

2.2 The Bain transformation path

The Bain transformation path describes a structural transition between bcc and fcc symmetries via intermediate tetragonal phases⁷⁵ of body-centered – or equivalently – face-centered tetragonal symmetry. Originally investigated for iron⁷⁵, the Bain path is relevant in thermo-mechanical processing – a central aspect for steel properties⁷⁶ – as it serves as a model for temperature-induced transitions between fcc (γ) and bcc (α) iron⁷⁷. The Bain path is also crucial for understanding properties of epitaxial films^{78,79} or metal nanowires⁸⁰.

Practically, the structures constituting a Bain path can be obtained by varying the ratio c/a between lattice parameters a and c of a tetragonal structure (cf. Fig. S1a); $c/a = 1$ corresponds to a cubic structure. We generate tetragonal geometries for lattice parameters a, c taking values in $[3.0\text{\AA}, 6.0\text{\AA}]$ with steps of 0.05\AA , resulting in 3721 crystal structures. These structures are then classified with ARISE, and the results depicted via classification and uncertainty maps in Fig. S1b and c, respectively. Each point in these maps corresponds to a prediction for a specific geometry. We include in the training set fcc, bcc, and tetragonal geometries with structural parameters known experimentally; they are shown as stars in Fig. S1b. Specifically, the lattice parameters ($a, c, c/a$) are $(3.155\text{\AA}, 3.155\text{\AA}, 1.0)$ for the bcc⁸¹ and $(3.615\text{\AA}, 5.112\text{\AA}, \sqrt{2})$ for the fcc prototype⁸², while two tetragonal structures (being assigned one common label “tetragonal”) are included with $(3.253\text{\AA}, 4.946\text{\AA}, 1.521)$ in case of In⁸³ and $(3.932\text{\AA}, 3.238\text{\AA}, 0.824)$ for α – Pa⁸⁴. We isotropically scale every geometry to remove one degree of freedom (see Method section), so that all possible cubic lattices are effectively equivalent; this allows the model to generalize by construction to all cubic lattices regardless of the lattice parameter. The same holds for tetragonal structures (i.e., two degrees of freedom) with constant c/a ratio. As visual aid, we mark lines of constant c/a in Fig. S1b-c starting from the four structures included in the training set. Note that any path connecting the constant c/a ratios corresponding to fcc and bcc structures constitutes a Bain path. To obtain a classification label, we select the class with the higher classification probability through an *argmax* operation (i.e., the label c maximizing Eq. 4). These predictions are shown in Fig. S1.

The model is able to detect the bcc and fcc phases in the expected areas, while all prototypes not being fcc, bcc, or tetragonal are correctly labeled as “Other”. We point out that only four structures – corresponding to points in the plot marked by the four stars – are included in the training set, while all other 3717 structures are model (test) predictions. We can also observe that the model correctly predicts the presence of a tetragonal phase between fcc (yellow band) and bcc (green band), even though no tetragonal structures from this region are included in the training set. This transition is smooth, only interrupted by small areas for which other, low-symmetry prototypes are assigned, but with high uncertainty, as quantified by the mutual information, cf. Fig. S1 c. We provide the classification probabilities of all assigned prototypes in Fig.

S3. In general, increased uncertainty appears at transitions between the assignments of different prototypes. We also note that there is a smooth transition for classification probabilities at the transition between prototypes (cf. Fig. S3). These results represent a first indication that the network has learned physically meaningful representations. Surprisingly, for large or small c/a ratios, i.e., points far outside the training set, other (low-symmetry) phases appear such as base-centered orthorhombic molecular iodine or face-centered orthorhombic γ –Pu with small uncertainty. While it may be desirable to avoid overconfident predictions far away from the training set, the assignments could be actually physically justified given the similarities between tetragonal and orthorhombic lattices, the most evident being that all angles in both crystal systems are 90° . We note that the transition to these prototypes is encompassed by regions of high uncertainty also in this case in agreement with physical intuition.

2.3 When the model is forced to fail: analysis of ARISE out-of-sample predictions

To assess the physical content learned by the network, we investigate its predictions – and thus its generalization ability – on structures corresponding to prototypes not included in the training. This is of particular relevance if one wants to use predictions of ARISE – for applications such as screening of large databases, or create low-dimensional maps for a vast collection of materials⁸⁵.

Given an unknown structure, the network needs to decide – among the classes it has been trained on – which one is the most suitable. It will assign the most similar prototypes and quantify the similarity via classification probabilities, providing a ranking of candidate prototypes. The uncertainty in the assignment, as quantified by mutual information, measures the reliability of the prediction. Note that the task of assigning the most similar prototype(s) to a given structure among 108 possible classes (and quantifying the similarity) is a very complicated task even for trained materials scientists, in particular in case of complex periodic and possibly defective three-dimensional structures.

We consider three examples (cf. Fig. S2 left): fluorite and tungsten Carbide (from AFLOW) where the correct labels are known, and one structure from the NOMAD encyclopedia (see last paragraph of this section for details on the provenance), for which the assigned space group is 1, i.e., no symmetry can be identified (via spglib). In all three cases there is no prototype in the dataset which represent a match for any of these structures. This is on purpose: the network will “fail” by construction since the correct class is not included in the possible classes the network knows (and needs to choose from). Analyzing how the network fails will give us insight on the physical content of the learned model. This test can also be viewed as discovering “unexpected similarities” across materials of different chemical composition and dimensionality.

Following the pipeline for single-crystal classification sum-

	Pristine	Random displacements (δ)					Missing atoms (η)			
		0.1%	0.6%	1%	2%	4%	1%	5%	10%	20%
Spglib (loose)	100.00	100.00	100.00	95.26	0.20	0.00	11.23	0.00	0.00	0.00
Spglib* (loose)	67.71	67.71	67.71	65.83	14.51	0.00	15.73	0.03	0.00	0.00
Spglib (tight)	100.00	0.00	0.00	0.00	0.00	0.00	11.23	0.00	0.00	0.00
Spglib* (tight)	83.33	0.00	0.00	0.00	0.00	0.00	17.53	0.00	0.00	0.00
PTM	100.00	100.00	100.00	100.00	100.00	100.00	88.67	51.76	25.93	6.24
PTM*	8.78	11.37	11.37	11.37	11.37	11.37	10.08	5.90	2.96	0.71
CNA	66.14	62.81	62.81	54.55	32.34	31.41	55.86	32.50	15.75	3.07
CNA*	1.44	1.62	1.62	1.40	0.83	0.81	1.44	0.84	0.41	0.08
a-CNA	100.0	100.0	100.0	100.0	100.0	100.0	89.25	52.81	25.92	5.37
a-CNA*	2.49	3.08	3.08	3.08	3.08	3.08	2.75	1.64	0.81	0.17
BAA	100.0	100.0	100.0	100.0	100.0	97.85	99.71	88.78	65.21	25.38
BAA*	2.49	3.08	3.08	3.08	3.08	3.03	3.08	2.74	2.02	0.81
ARISE (108 / 108)	100.00	100.00	100.00	100.00	99.86	99.29	100.00	100.00	100.00	99.85

Table S1. Accuracy in identifying the parent class of defective crystal structures. Two lines are shown for each of the methods used for benchmarking (spglib, PTM, CNA, a-CNA, BAA): In rows without stars, the accuracy is calculated only for structures for which the respective method was designed for; for instance, spglib can be applied to every structure of Fig. 1e except the 12 nanotubes (note that we only include prototypes from AFLOW for spglib, cf. S2.1). This is also true for the other methods, while additional structures have to be removed for instance for CNA, a-CNA, and BAA as they cannot classify simple cubic and diamond structures. In starred rows, all 108 classes summarized in Fig. 1e are included, leading to the strong decrease in performance. In contrast, the neural network approach proposed here can be applied to all classes, and thus the whole dataset was used.

	Random displacements (δ)		Missing atoms (η)	
	7%	10%	25%	30%
Spglib (loose)	0.00	0.00	0.00	0.00
Spglib* (loose)	0.00	0.00	0.00	0.00
Spglib (tight)	0.00	0.00	0.00	0.00
Spglib* (tight)	0.00	0.00	0.00	0.00
PTM	100.00	94.34	3.33	1.72
PTM*	11.37	10.71	0.38	0.19
CNA	31.41	24.20	1.38	0.55
CNA*	0.81	0.62	0.04	0.01
a-CNA	99.99	94.55	2.60	1.03
a-CNA*	3.08	2.90	0.08	0.03
BAA	87.79	69.68	14.25	7.35
BAA*	2.77	2.22	0.49	0.30
ARISE (this work)	97.82	94.56	99.86	99.76

Table S2. Accuracy in identifying the parent class of defective crystal structures for high displacements (percentage δ) and missing atoms (percentage η).

	Missing atoms and displacements (η , δ)						
	(1%, 0.1%)	(5%, 0.6%)	(10%, 1%)	(15%, 2%)	(20%, 4%)	(25%, 7%)	(30%, 10%)
Spplib (loose)	11.32	0.00	0.00	0.00	0.00	0.00	0.00
Spplib* (loose)	15.76	0.00	0.00	0.00	0.00	0.00	0.00
Spplib (tight)	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Spplib* (tight)	0.00	0.00	0.00	0.00	0.00	0.00	0.00
PTM	88.68	51.78	25.60	12.75	6.41	3.19	1.46
PTM*	10.08	5.90	2.92	1.45	0.73	0.36	0.16
CNA	55.77	31.95	13.83	4.41	2.03	0.79	0.19
CNA*	1.44	0.82	0.36	0.11	0.05	0.02	0.00
a-CNA	89.21	52.36	26.01	12.13	6.07	2.40	0.97
a-CNA*	2.75	1.62	0.81	0.38	0.19	0.08	0.03
BAA	99.72	88.98	65.17	42.62	25.95	15.58	6.63
BAA*	3.07	2.75	2.02	1.34	0.82	0.50	0.22
ARISE (this work)	100.00	100.00	100.00	99.88	99.29	97.31	92.50

Table S3. Accuracy in identifying the parent class of defective crystal structures, with both missing atoms (percentage η) and displacements (percentage δ) introduced at the same time. The results show that ARISE is also robust for highly defective structures where displacements and missing atoms are present at the same time. This is the typical situation encountered in APT – making ARISE arguably the best available candidate for crystal structure classification in APT experimental data.

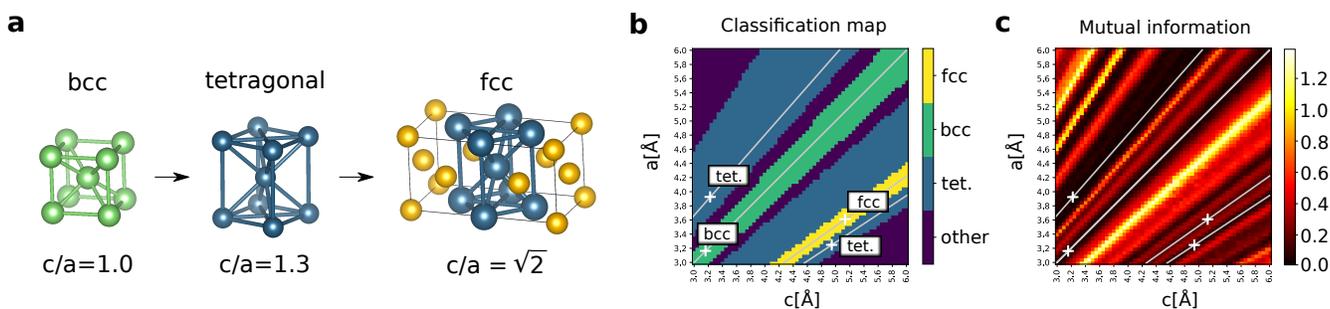


Figure S1. Application of ARISE for the characterization of the Bain path. **a** Structures occurring in the Bain path, obtained by varying c/a ; increasing c/a from 1.0 (bcc) leads to transitions to tetragonal phases and finally to the fcc structure ($c/a=\sqrt{2}$). **b** Classification (argmax predictions, left) and uncertainty (mutual information, right) for geometries in the range $c, a \in [3.0\text{\AA}, 6.0\text{\AA}]$. Geometries included in the training set are marked by stars in **b,c**. As we isotropically scale the structures, geometries with constant c/a are equivalent, which is indicated by solid lines.

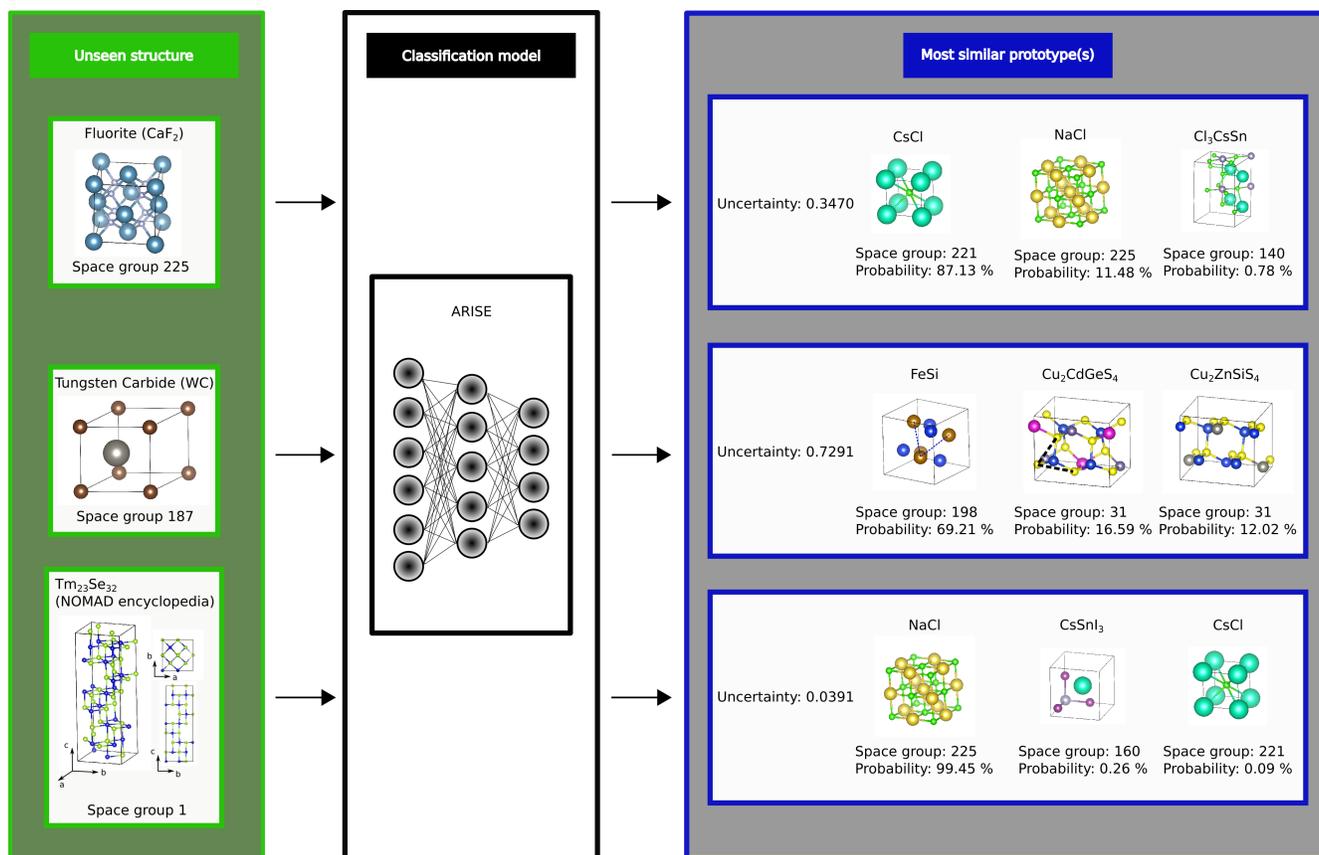


Figure S2. Three examples for assigning the most similar prototype(s) (right panel) to structures for which the corresponding structural class is not contained in the training set of ARISE (left panel). For each prediction, space group and classification probabilities of the top predictions are specified together with an uncertainty estimate (mutual information). The space groups are returned via spglib, where we choose the highest symmetry that is found for all combinations of precision parameters (0.1, 0.01, 0.001, 0.0001) [Å] and angle tolerances (1, 2, 3, 4, 5) [°].

marized in Fig. 1, we compute classification probabilities and mutual information, yielding the assignments shown in Fig. S2 right. To rationalize the predictions shown in Fig. S2 from a physical standpoint, we inspect the substructures formed by the chemical species in both original and assigned structures. This is motivated by our choice of materials representation as averaged SOAP descriptor of substructures (see Methods for more details). The two most similar prototypes to fluorite (CaF₂) are CsCl and NaCl, both consisting of two inter-penetrating lattices of the same type, two sc lattices for CsCl and two fcc lattices for NaCl. Fluorite contains both sc (F atoms) and fcc (Ca atoms) which is likely why CsCl and NaCl are assigned, together with a ternary halide tetragonal perovskite, also containing sc symmetry (via Cs and Sn atoms, respectively).

For tungsten carbide (WC), T and C form two hexagonal lattices. In the unit cell of the most similar prototype, FeSi, 60° angles are formed within the substructures of each species (see dashed lines in the unit cell), thus justifying this classification. Furthermore, two quaternary chalcogenides appear as further candidates. This similarity – hard to assess by

eye – originates by the presence of angles close to 60° for S atoms (yellow) for both Cu₂CdGeS₄ and Cu₂ZnSiS₄ (marked in the figure for Cu₂CdGeS₄). Also note that these two quaternary prototypes, Cu₂ZnSiS₄ and Cu₂CdGeS₄ are a result of substituting Ge and Si with isoelectric elements Zn and Cd, which implies that these structures are expected to be similar. This explains why they both appear as candidates for structures being similar to tungsten carbide. Finally, for the compound Tm₂₃Se₃₂ from the NOMAD encyclopedia, the model identifies NaCl as the most similar prototype. Looking at the structure from different angles, especially from the top (cf. Fig. S2, left part), a similarity to cubic systems can be identified. The classification method robustness to missing atoms makes the apparent gaps in the side-view negligible, and thus rationalizes the NaCl assignment. Regarding the uncertainty quantification (via mutual information), increased uncertainties appear for fluorite and tungsten carbide, since besides the top prediction with more than 70% classification probability, other prototypes are possible candidates for the most similar prototype. For the NOMAD structure Tm₂₃Se₃₂, the network is quite confident, most likely because no other

good candidates are presented among the binaries included in the 108 classes dataset.

These results show that the model – even when forced to fail by construction – returns (highly non-trivial) physically meaningful predictions. This makes ARISE particularly attractive for screening large and structurally diverse databases, in particular assessing structures for which no symmetry label can be obtained with any of the current state-of-the-art methods.

The structure taken from the NOMAD encyclopedia can be found at <https://encyclopedia.nomad-coe.eu/gui/#/material/249163> and has the ID mp-684691 in materials project. It is a derived structure from Tm₂Se₃ (ID = mp-1189318) and TmSe (ID = mp-2822), which can be found at <https://materialsproject.org/materials/ID/#snl> where ICSD IDs, i.e., the experimental origin of the respective structures are provided.

In addition to the analysis in the main text (Fig. S2), we present some results for further out-of-sample structures:

- Boron nitride (bulk, graphitic, http://aflowlib.org/CrystalDatabase/AB_hP4_194_c_d.html) classified as hexagonal graphite (probability 63.32%), mut.inf. 0.7278
- Cementite (http://aflowlib.org/CrystalDatabase/AB3_oP16_62_c_cd.html) classified as MnP with probability 49.14%, mut.inf. 0.7176
- CuTi₃ (L60 Strukturbericht, http://aflowlib.org/CrystalDatabase/AB3_tP4_123_a_ce.html) classified as bct α -Pa with probability 78.41%, mut.inf. 0.8539
- Benzene (http://aflowlib.org/CrystalDatabase/AB_oP48_61_3c_3c.html) classified as nanotube (chiral indices (n,m)=(5,2)) with probability 68.48%, mut.inf. 0.6249
- Rutile (http://aflowlib.org/CrystalDatabase/A2B_tP6_136_f_a.html) classified as orthorhombic halide perovskite with probability 44.62%, mut.inf. 0.8733
- NbO (http://aflowlib.org/CrystalDatabase/AB_cP6_221_c_d.html), which is NaCl with 25% ordered vacancies on both the Na and Cl sites, classified as NaCl with probability 99.96%, mut.inf. 0.0027
- Moissanite 4H SiC (http://aflowlib.org/CrystalDatabase/AB_hP8_186_ab_ab.html) classified as wurtzite with probability 99.74%, mut.inf. 0.0166
- K₂PtCl₆ (http://aflowlib.org/CrystalDatabase/A6B2C_cF36_225_e_

[c_a.html](#)) classified as NaCl with probability 61.4%, mut.inf. 0.5402

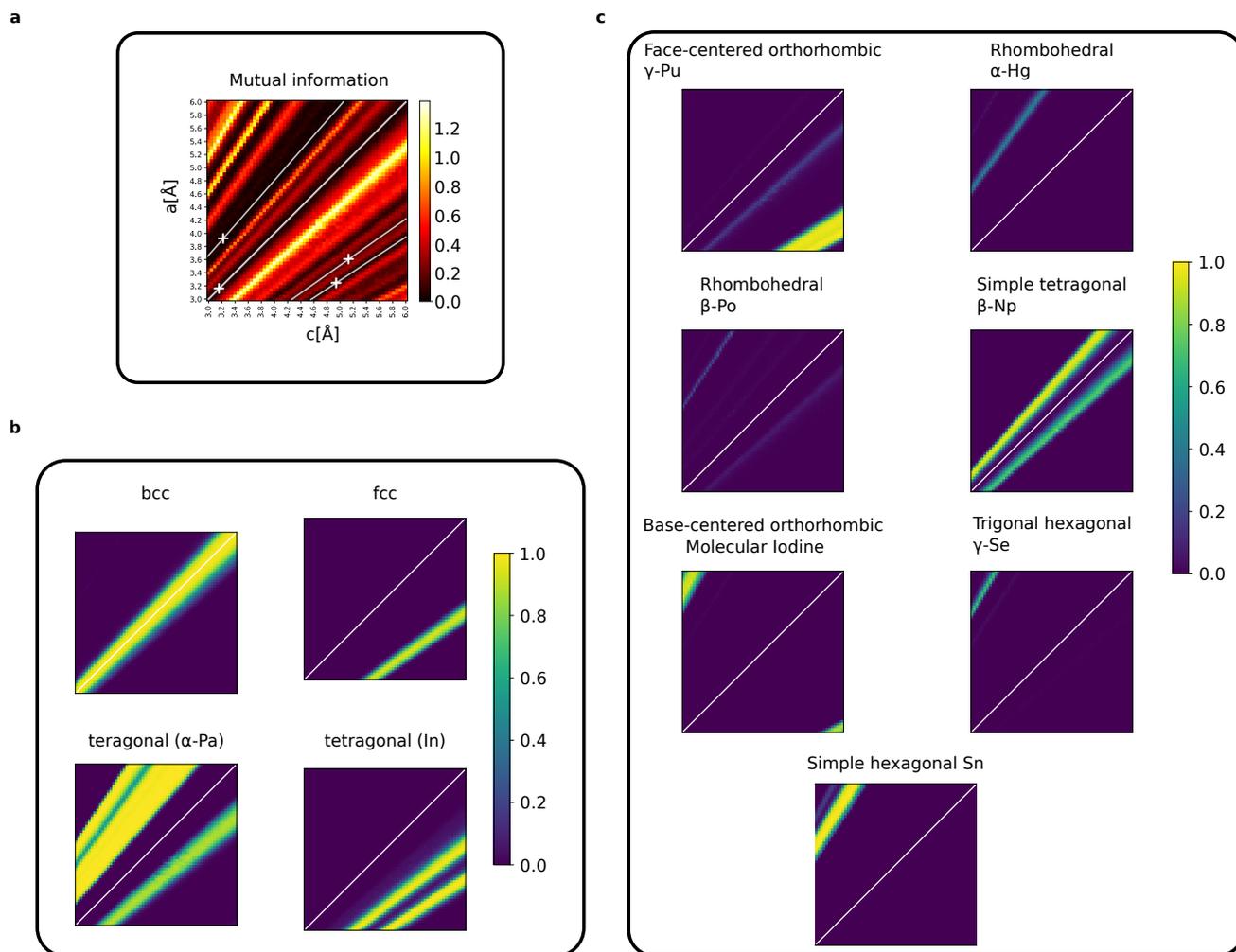


Figure S3. Bain path - all prototypes with increased classification probability: **a** Mutual information plot showing the spots of high and low uncertainty for different geometries. **b** Classification probability maps corresponding to bcc, fcc and two tetragonal phases. **c** Representative selection of other prototypes showing non-zero classification probabilities.

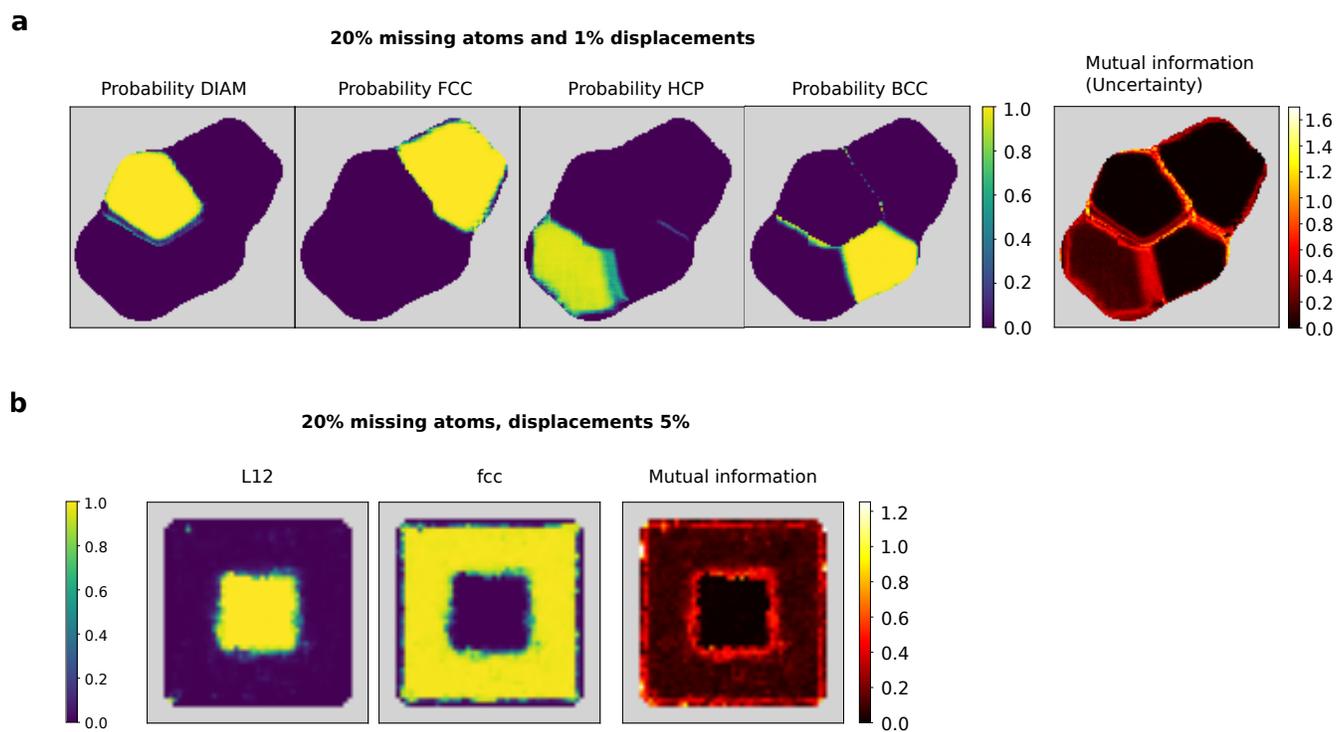


Figure S4. Investigation of distorted synthetic polycrystals. **a** Analysis on defective version of mono-species polycrystal shown in Fig. 2a. **b** Analysis on defective version of superalloy structure shown in Fig. 2h.

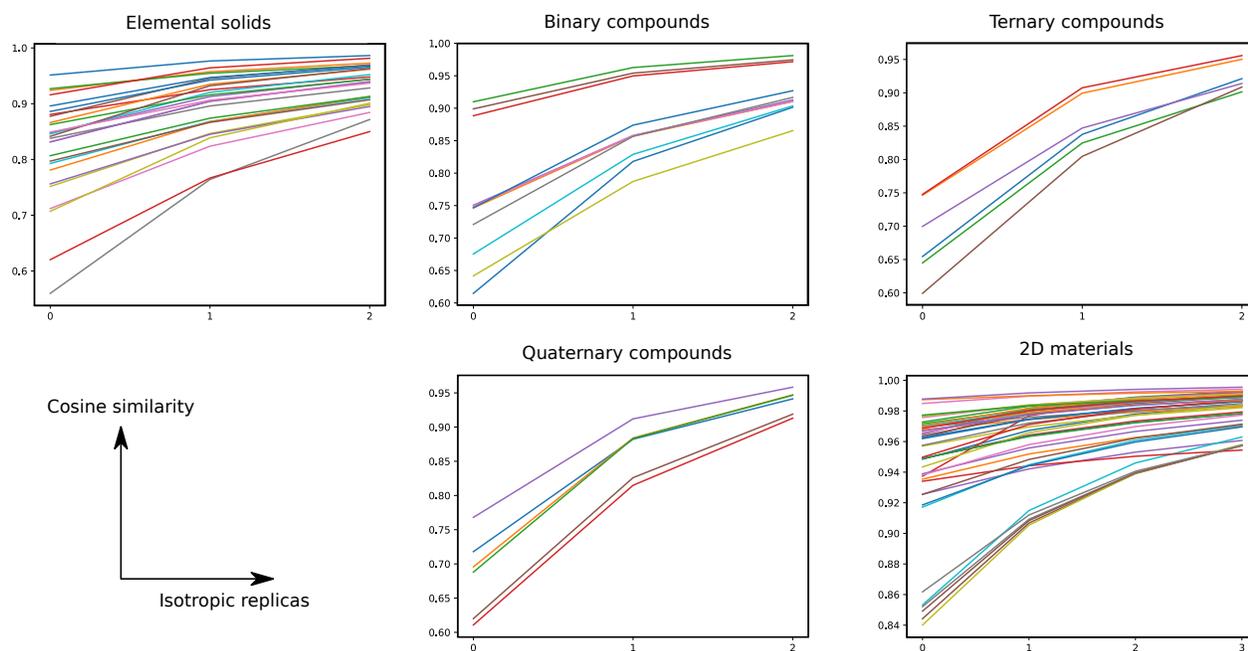


Figure S5. Cosine similarity plots for elemental, binary, ternary, and quaternary compounds as well as 2D materials (for SOAP settings $R_C = 4.0 \cdot d_{NN}$, $\sigma = 0.1 \cdot d_{NN}$, and $exsf = 1.0$ corresponding to the center of the parameter range used in the training set). Each line corresponds to a particular prototype. The x-axis corresponds to three different (non-periodic) supercells, where supercell “0” stands for the smallest isotropic supercell (for instance $4 \times 4 \times 4$ repetitions) for which at least 100 atoms are obtained. Supercells “1” and “2” correspond to the next two bigger isotropic replicas (e.g., $5 \times 5 \times 5$ and $6 \times 6 \times 6$). The y-axis corresponds to the cosine similarity of the respective supercell to the periodic structure, i.e., the case of infinite replicas. One can see a continuous increase of similarity with larger supercell size, where for the largest supercell, the similarity is greater than 0.8 for all prototypes. Thus, it is to be expected that systems sizes larger than those included in the training set can be correctly classified by ARISE. For smaller systems, however, generalization ability will depend on the prototype. Practically, one can include smaller supercells in the training set, which is not a major problem due to fast convergence time.

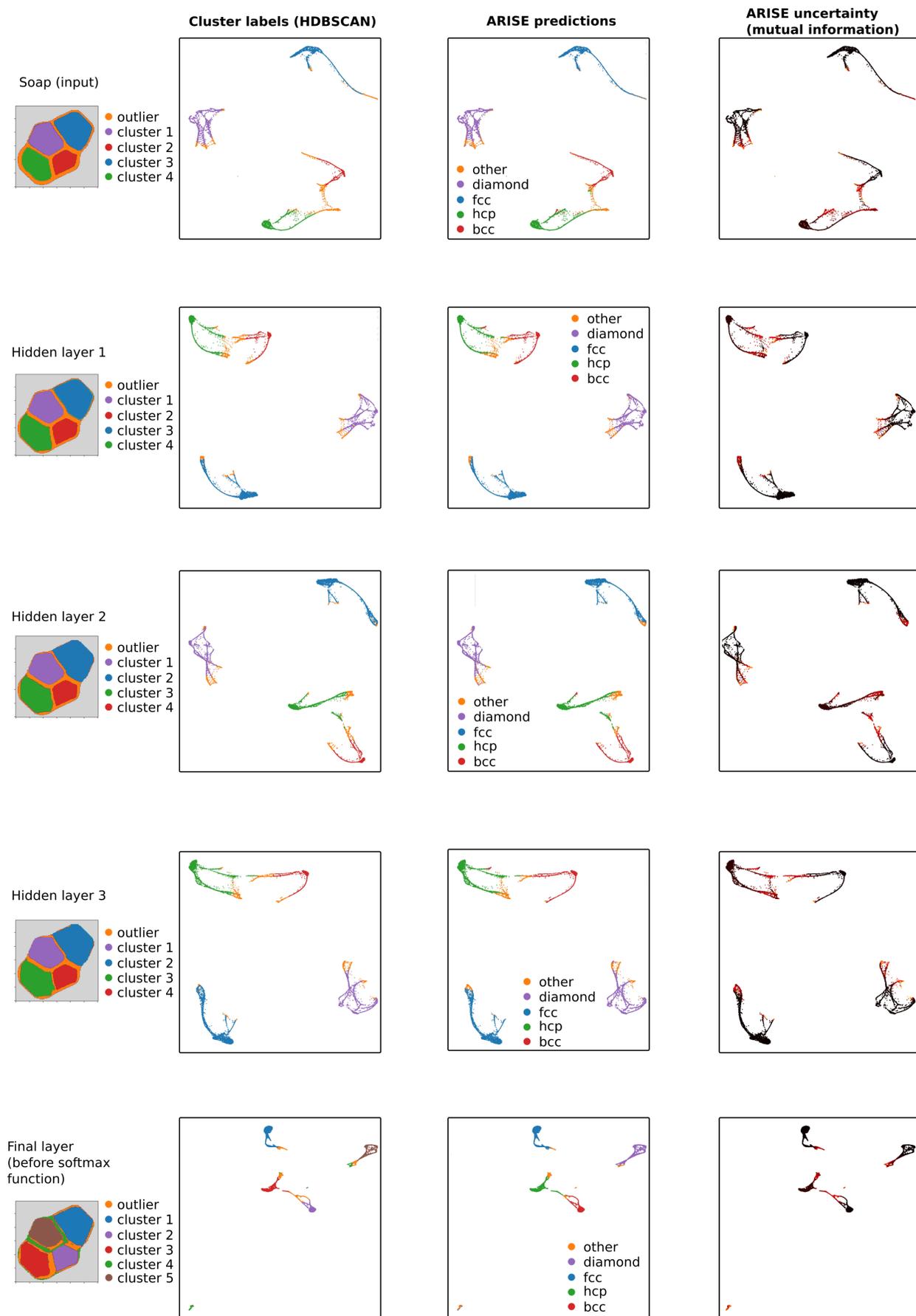


Figure S6. Unsupervised analysis analogous to Figure 2d-e, for all layers (before the ReLU or rather the softmax function).

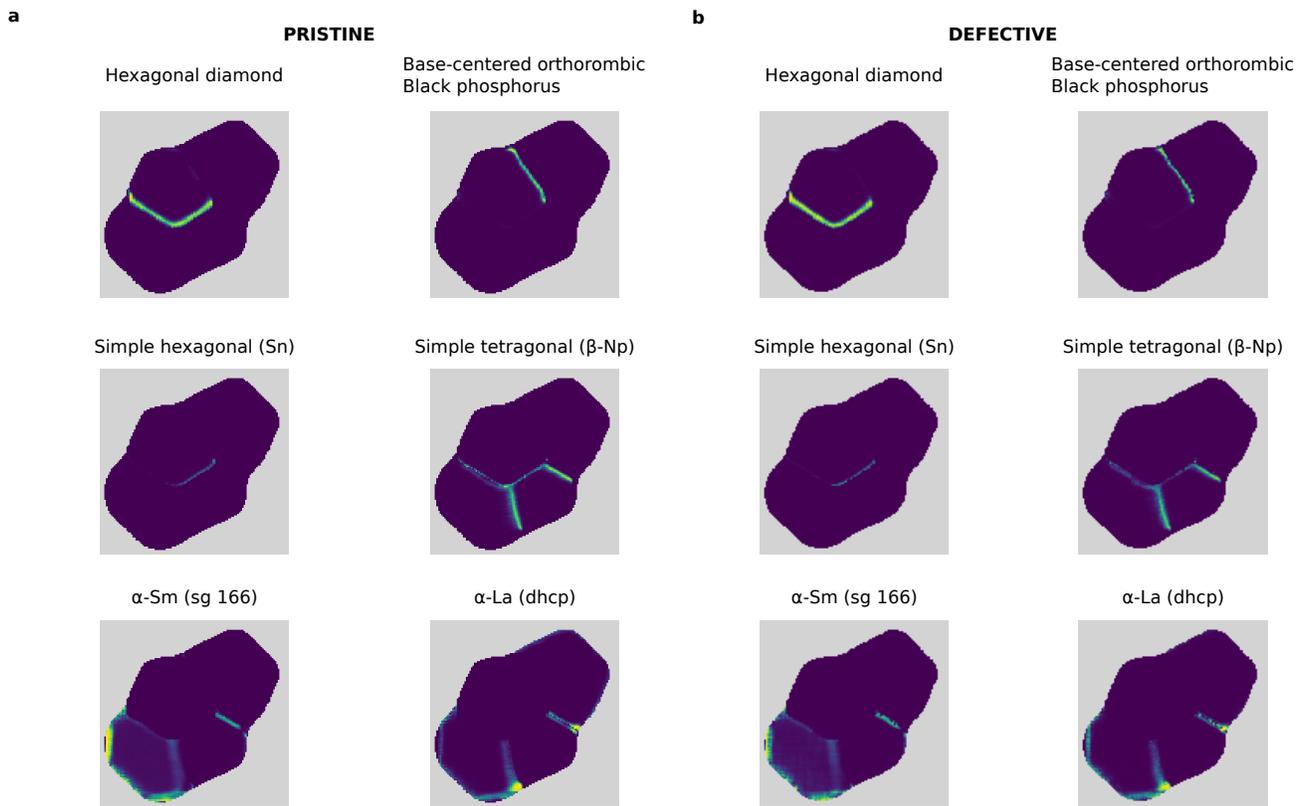


Figure S7. Probability maps of the most important prototypes for both pristine (**a**) and defective (**b**) version of the mono-species polycrystal in Fig. 2a.

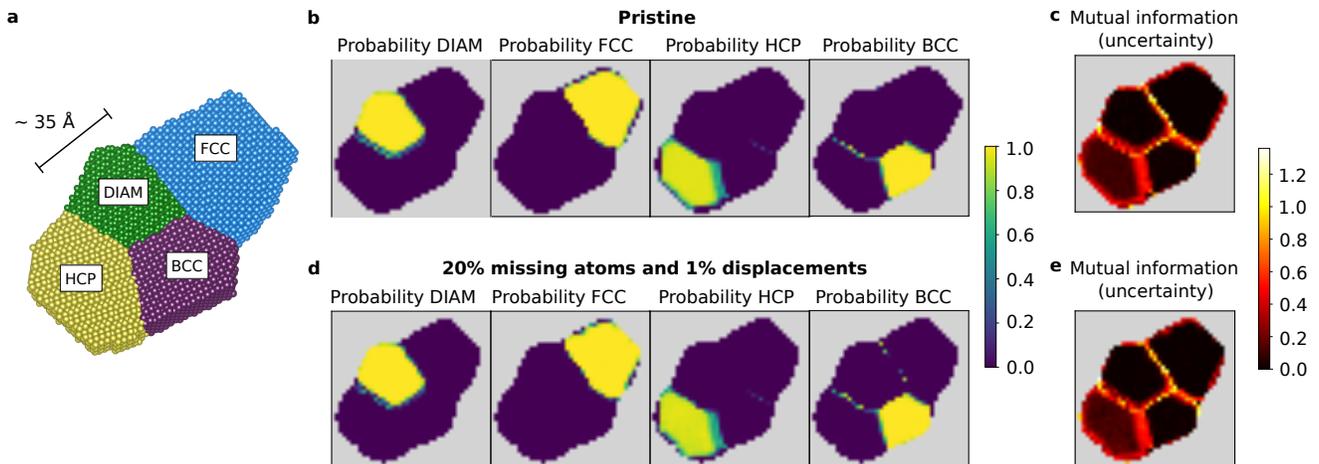


Figure S8. Mono-species elemental polycrystal investigation via strided pattern matching using lower resolution (stride of 3.0 Å in both x and y direction opposed to 1.0 Å as in Fig. 2). Choosing the stride is a trade-off between computation time and resolution. For instance, at the grain boundary between diamond and hcp, the transition from diamond to hexagonal diamond to hcp (cf. Fig. S7) are recognized in Fig. 2b, while being obscured in the presented low resolution pictures.

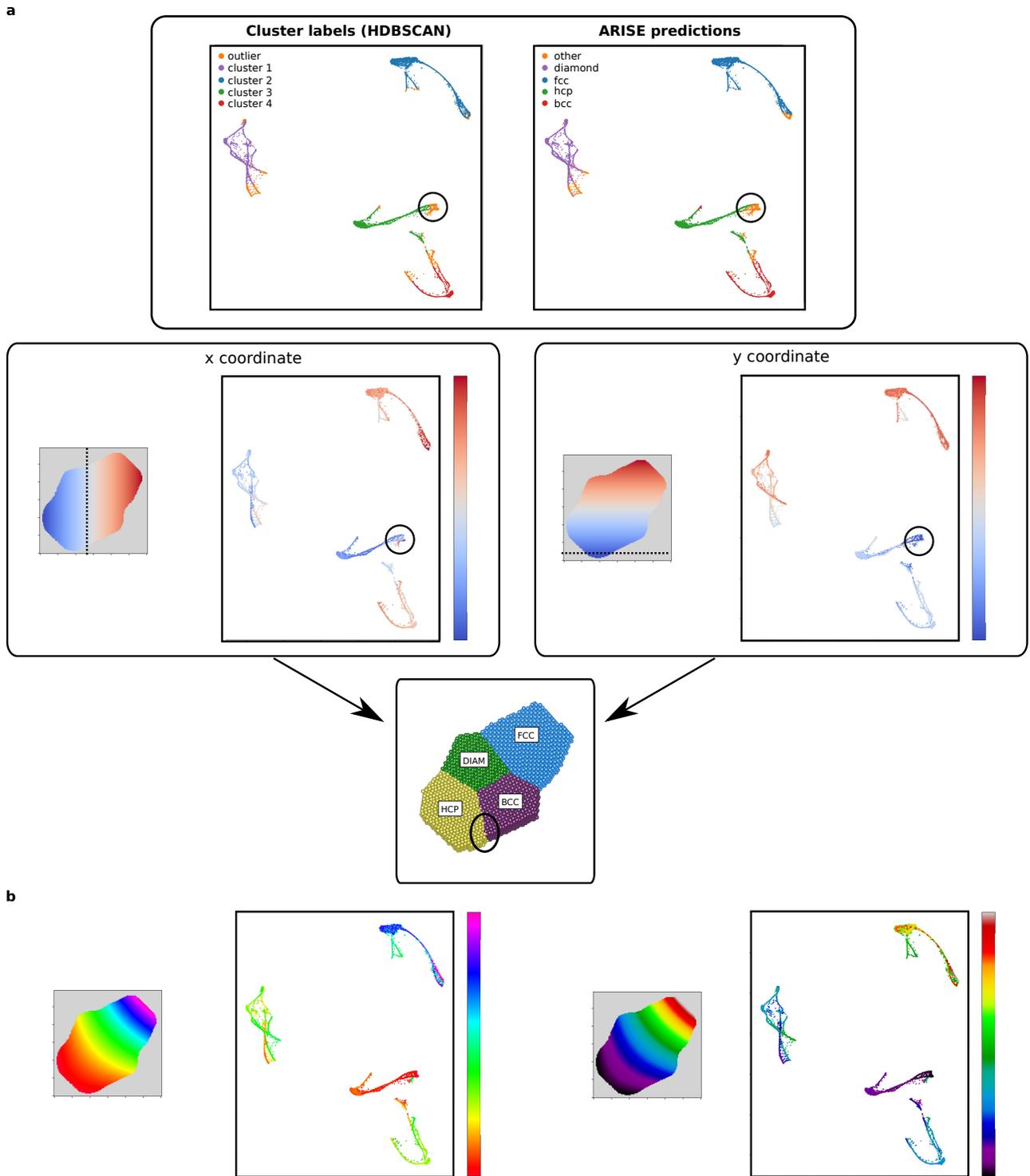


Figure S9. Grain boundary mapping in 2D UMAP projections (here for the second hidden layer, before the ReLU function). This figure confirms the observation that ARISE’s representations of different types of grain boundaries (here: transitions between fcc, bcc, hcp, and diamond) are mapped to different regions in the UMAP projection. **a** As a guideline, the top part shows the UMAP projections discussed in the main text, which are colored by the HDBSCAN cluster assignments (left) and ARISE’s predictions (right). We exemplarily focus on the circled region. In the central panels, crystal maps and UMAP projections are colored according to the x and y positions of the local boxes that correspond to each pixel. As visualized by the color code and the dashed lines in the crystal maps, x and y positions of this grain boundary correspond to the transition region marked in the bottom panel. Similarly, one may proceed for other regions. **b** Different color codes visualizing the way in which crystal regions (e.g., grain boundaries) appear in the UMAP projection.

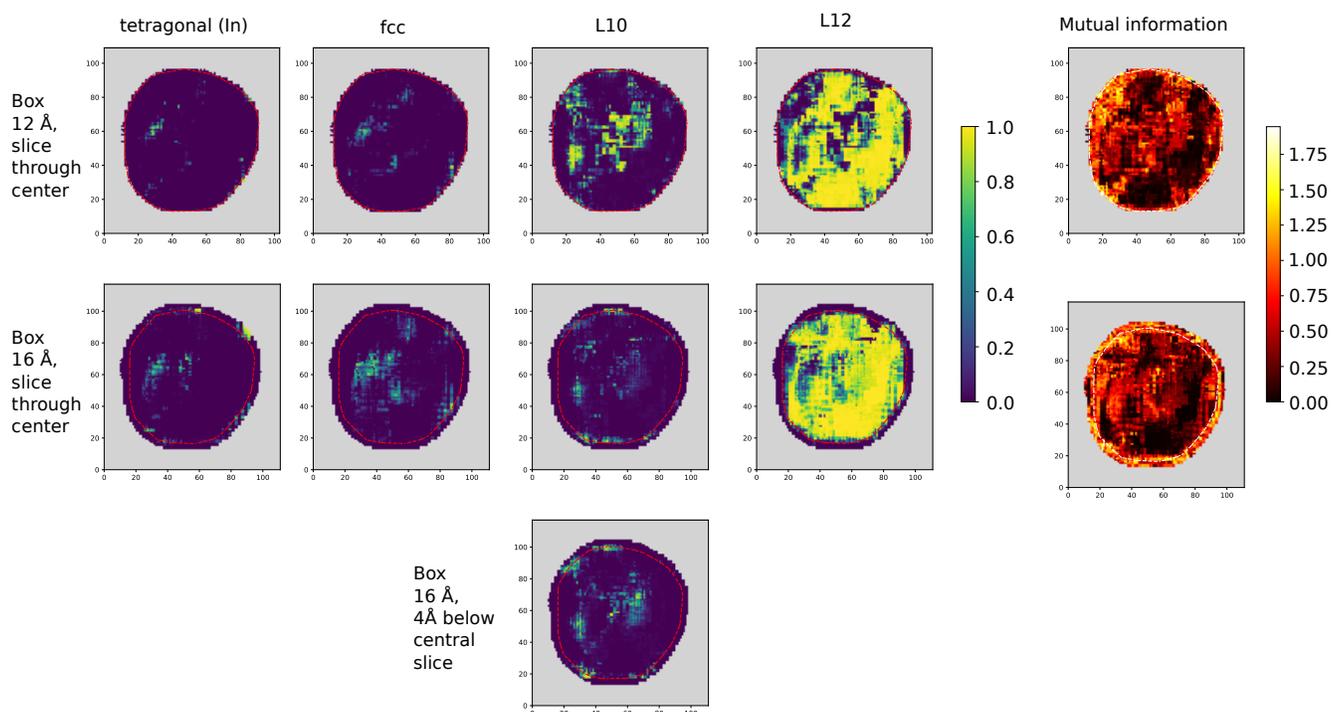


Figure S10. Comparison of crystal maps (slice through center, most important prototypes and mutual information) for AET nanoparticle data⁶⁰ for two different box sizes. Dashed lines indicate the crystal boundaries in all 2D maps. ARISE allows to detect the appearance of the tetragonally distorted fcc prototype (In). For larger box sizes, the fcc assignment increases in the center and also the L1₂ classification probability rises. While the central slice of the L1₀ prototype for a box size of 16 Å shows only weak signal, a slice slightly below reveals higher probability (see bottom, isolated slice), i.e., ARISE does not overlook this physically relevant phase.

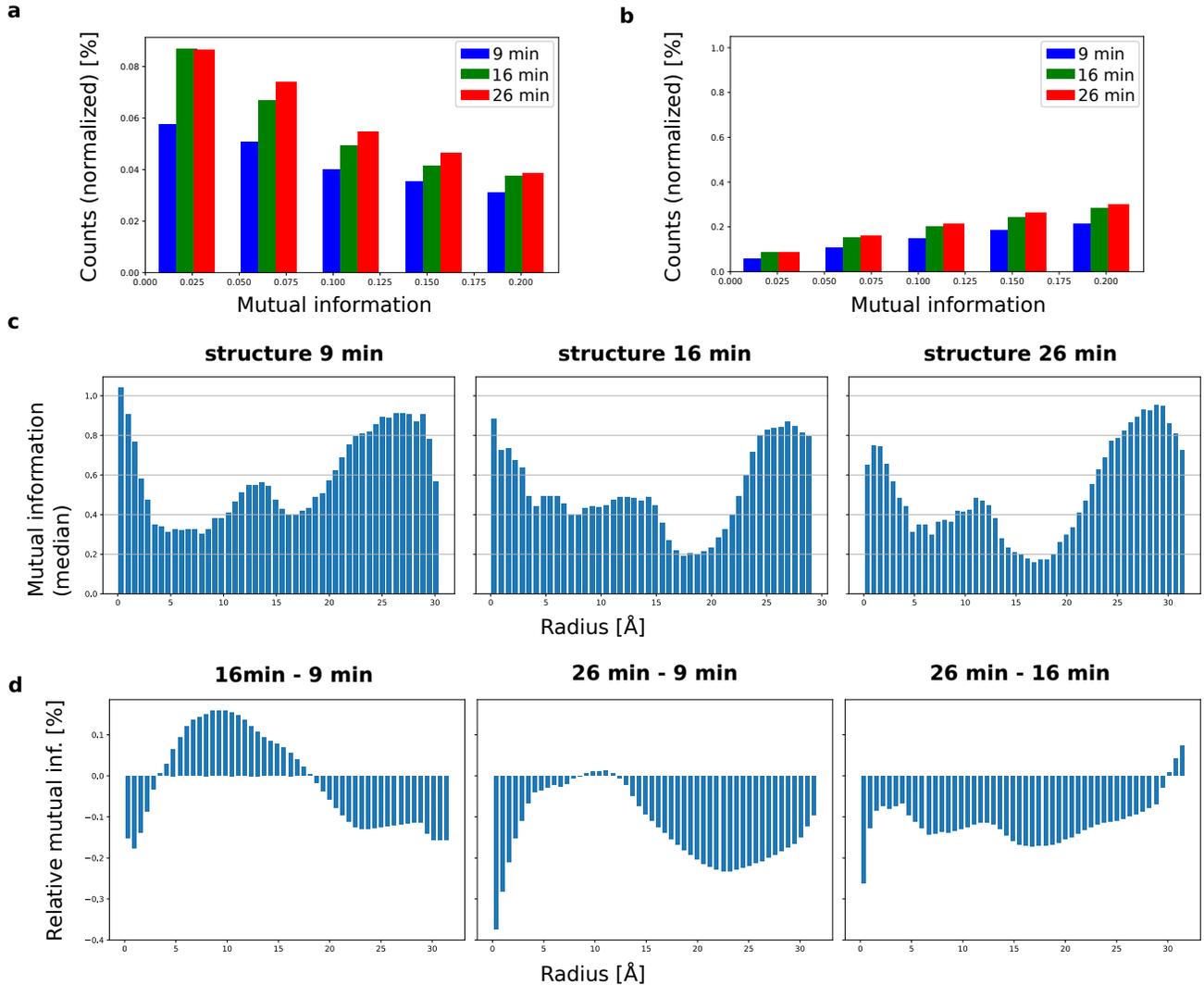


Figure S11. Quantitative study of mutual information distribution for different annealing times. **a** Histogram of mutual information values for each annealing time (where the corresponding histograms are normalized via dividing each bin by the total number of boxes). Only mutual-information values smaller than 0.2 are shown, which correspond to the "dark", i.e., low mutual information spots in Fig. 4c. **b** Cumulative distribution calculated for the histogram shown in **a**. From **a**, **b** it is apparent that the number of low-uncertainty boxes increases for larger annealing times. **c-d** Investigation of the radial distribution of the mutual information. **c** Histograms of uncertainty (mutual information) obtained via spatially binning the SPM maps of 4c into spherical shells, where the median is computed for each bin. Given a mutual information value, the associated radius is calculated as the distance of the center of the corresponding box (as obtained via SPM) to the center of the most central box. **d** Each panel shows the difference between the cumulative distributions of two annealing times, where the cumulative distributions are calculated from the histograms shown in **b**. In addition the histograms are normalized the following way: Given the times t_1, t_2 with $t_1 < t_2$, the cumulative sum of $t_2 - t_1$ is calculated and then divided by the cumulative sum of time t_1 such that the fractional change from t_1 to t_2 is obtained. One can conclude that in **c** a clear decrease of mutual information can be spotted in specific regions, e.g., for the radial region 15-20 Å. The cumulative sums that are used in **d** allow to quantify the order more globally in the sense that each bin (of the cumulative sum corresponding to a specific annealing time) is proportional to the spherically averaged integral from radius zero up to the radius corresponding to the bin. Since the particle sizes are changing over time due to diffusion, the particles have different size. Thus, we single out a radius at which to compare the global order: for instance comparing the bins corresponding to a radius of $r=25$ Å, we see that for all three panels, the values are negative and thus the structure that has been annealed longer shows larger global order.

#	Prototype	Symmetry	Material type	Data source
1.	bcc (W)	229, cubic	Bulk, Elemental solid	AFLOW / NOMAD
2.	diamond (C)	227, cubic	Bulk, Elemental solid	AFLOW / NOMAD
3.	fcc (Cu)	225, cubic	Bulk, Elemental solid	AFLOW / NOMAD
4.	α -Po	221, (simple) cubic	Bulk, Elemental solid	AFLOW / NOMAD
5.	hcp (Mn)	194, hexagonal	Bulk, Elemental solid	AFLOW / NOMAD
6.	α -La (dhcp)	194, hexagonal	Bulk, Elemental solid	AFLOW / NOMAD
7.	Hexagonal diamond	194, hexagonal	Bulk, Elemental solid	AFLOW / NOMAD
8.	Hexagonal graphite	194, hexagonal	Bulk, Elemental solid	AFLOW / NOMAD
9.	Sn	191, (simple) hexagonal	Bulk, Elemental solid	AFLOW / NOMAD
10.	Buckled graphite	186, hexagonal	Bulk, Elemental solid	AFLOW / NOMAD
11.	α -As	166, rhombohedral	Bulk, Elemental solid	AFLOW / NOMAD
12.	α -Hg	166, rhombohedral	Bulk, Elemental solid	AFLOW / NOMAD
13.	α -Sm	166, rhombohedral	Bulk, Elemental solid	AFLOW / NOMAD
14.	β -O	166, rhombohedral	Bulk, Elemental solid	AFLOW / NOMAD
15.	β -Po	166, rhombohedral	Bulk, Elemental solid	AFLOW / NOMAD
16.	γ -Se	152, trigonal hexagonal	Bulk, Elemental solid	AFLOW / NOMAD
17.	Rhombohedral graphite	166, rhombohedral	Bulk, Elemental solid	AFLOW / NOMAD
18.	α -Pa	139, (body-centered) tetragonal	Bulk, Elemental solid	AFLOW / NOMAD
19.	β -Sn	141, (body-centered) tetragonal	Bulk, Elemental solid	AFLOW / NOMAD
20.	In	139, (body-centered) tetragonal	Bulk, Elemental solid	AFLOW / NOMAD
21.	γ -N	136, (simple) tetragonal	Bulk, Elemental solid	AFLOW / NOMAD
22.	β -Np	129, (simple) tetragonal	Bulk, Elemental solid	AFLOW / NOMAD
23.	γ -Pu	70, (face-centered) orthorhombic	Bulk, Elemental solid	AFLOW / NOMAD
24.	α -Ga	64, (base-centered) orthorhombic	Bulk, Elemental solid	AFLOW / NOMAD
25.	Black phosphorus	64, (base-centered) orthorhombic	Bulk, Elemental solid	AFLOW / NOMAD
26.	Molecular iodine	64, (base-centered) orthorhombic	Bulk, Elemental solid	AFLOW / NOMAD
27.	α -U	63, (base-centered) orthorhombic	Bulk, Elemental solid	AFLOW / NOMAD
28.	NaCl	225, cubic	Bulk, Binary compound	AFLOW / NOMAD
29.	CsCl	221, cubic	Bulk, Binary compound	AFLOW / NOMAD
30.	L1 ₂ (Cu ₃ Au)	221 (simple) cubic	Bulk, Binary compound	AFLOW / NOMAD
31.	Zinc blende (ZnS)	216, (face-centered) cubic	Bulk, Binary compound	AFLOW / NOMAD
32.	FeSi	198 (simple) cubic	Bulk, Binary compound	AFLOW / NOMAD
33.	NiAs	194, hexagonal	Bulk, Binary compound	AFLOW / NOMAD
34.	Wurtzite (ZnS)	186, hexagonal	Bulk, Binary compound	AFLOW / NOMAD
35.	L1 ₀ (CuAu)	123, (simple) tetragonal	Bulk, Binary compound	AFLOW / NOMAD
36.	CrB	63, (base-centered) orthorhombic	Bulk, Binary compound	AFLOW / NOMAD
37.	MnP	62, (simple) orthorhombic	Bulk, Binary compound	AFLOW / NOMAD
38.	FeB	62, (simple) orthorhombic	Bulk, Binary compound	AFLOW / NOMAD
39.	AgNbO ₃	cubic	Bulk, Ternary compound	CMR
40.	CsSnI ₃	cubic	Bulk, Ternary compound	CMR
41.	CsSnCl ₃	tetragonal	Bulk, Ternary compound	CMR
42.	Cs ₂ WO ₄	tetragonal	Bulk, Ternary compound	CMR
43.	Ca ₃ Ge ₂ O ₇	tetragonal	Bulk, Ternary compound	CMR
44.	CsSnCl ₃	orthorhombic	Bulk, Ternary compound	CMR

Table S4. Complete list of prototypes (part I) include in the training set of this work. If provided by the respective resources, information on space group, crystal system or Bravais lattice is listed.

#	Prototype	Symmetry	Material type	Data source
45.	Cu ₂ BaGeSe ₄	144 (trigonal)	Bulk, Quaternary compound	CMR
46.	Cu ₂ CdSnS ₄	121 (tetragonal)	Bulk, Quaternary compound	CMR
47.	Cu ₂ ZnSnS ₄	82 (tetragonal)	Bulk, Quaternary compound	CMR
48.	Cu ₂ KVS ₄	40 (orthorhombic)	Bulk, Quaternary compound	CMR
49.	Cu ₂ CdGeS ₄	31 (orthorhombic)	Bulk, Quaternary compound	CMR
50.	Cu ₂ ZnSiS ₄	7 (monoclinic)	Bulk, Quaternary compound	CMR
51.	Graphene	191 (hexagonal)	2D Materials	CMR
52.	Ti ₃ C ₂	187 (hexagonal)	2D Materials	CMR
53.	Ti ₃ C ₂ O ₂	187 (hexagonal)	2D Materials	CMR
54.	MoS ₂	187 (hexagonal)	2D Materials	CMR
55.	Ti ₃ C ₂ H ₂ O ₂	187 (hexagonal)	2D Materials	CMR
56.	GaS	187 (hexagonal)	2D Materials	CMR
57.	BN	187 (hexagonal)	2D Materials	CMR
58.	Ti ₂ CH ₂ O ₂	164 (trigonal)	2D Materials	CMR
59.	Ti ₂ CO ₂	164 (trigonal)	2D Materials	CMR
60.	CdI ₂	164 (trigonal)	2D Materials	CMR
61.	CH	164 (trigonal)	2D Materials	CMR
62.	CH ₂ Si	156 (trigonal)	2D Materials	CMR
63.	Ti ₄ C ₃	156 (trigonal)	2D Materials	CMR
64.	BiTeI	156 (trigonal)	2D Materials	CMR
65.	Ti ₄ C ₃ O ₂	156 (trigonal)	2D Materials	CMR
66.	GeSe	156 (trigonal)	2D Materials	CMR
67.	MoSSe	156 (trigonal)	2D Materials	CMR
68.	Ti ₄ C ₃ H ₂ O ₂	156 (trigonal)	2D Materials	CMR
69.	AgBr ₃	150 (trigonal)	2D Materials	CMR
70.	TiCl ₃	150 (trigonal)	2D Materials	CMR
71.	BiI ₃	147 (trigonal)	2D Materials	CMR
72.	FeSe	129 (tetragonal)	2D Materials	CMR
73.	PbSe	123 (tetragonal)	2D Materials	CMR
74.	GeS ₂	115 (tetragonal)	2D Materials	CMR
75.	C ₃ N	65 (orthorhombic)	2D Materials	CMR
76.	FeOCl	59 (orthorhombic)	2D Materials	CMR
77.	P	28 (orthorhombic)	2D Materials	CMR
78.	PdS ₂	14 (monoclinic)	2D Materials	CMR
79.	MnS ₂	14 (monoclinic)	2D Materials	CMR
80.	GaSe	12 (monoclinic)	2D Materials	CMR
81.	TiS ₃	11 (monoclinic)	2D Materials	CMR
82.	WTe ₂	11 (monoclinic)	2D Materials	CMR
83.	HfBrS	7 (monoclinic)	2D Materials	CMR
84.	RhO	6 (monoclinic)	2D Materials	CMR
85.	SnS	6 (monoclinic)	2D Materials	CMR
86.	NiSe	6 (monoclinic)	2D Materials	CMR
87.	AuSe	6 (monoclinic)	2D Materials	CMR

Table S5. Complete list of prototypes (part II) include in the training set of this work.

#	Prototype	Symmetry	Material type	Data source
88.	VTe ₃	6 (monoclinic)	2D Materials	CMR
89.	ReS ₂	2 (monoclinic)	2D Materials	CMR
90.	ScPSe ₃	1 (triclinic)	2D Materials	CMR
91.	PbA ₂ I ₄	1 (triclinic)	2D Materials	CMR
92.	PbS	1 (triclinic)	2D Materials	CMR
93.	CrW ₃ S ₈	1 (triclinic)	2D Materials	CMR
94.	VPSe ₃	1 (triclinic)	2D Materials	CMR
95.	CrWS ₄	1 (triclinic)	2D Materials	CMR
96.	MnPSe ₃	1 (triclinic)	2D Materials	CMR
97.	Carbon nanotube	armchair, (3,3), 30.0°, 4.07 Å	Nanotubes, mono-species	ASE
98.	Carbon nanotube	armchair, (4,4), 30.0°, 5.42 Å	Nanotubes, mono-species	ASE
99.	Carbon nanotube	chiral, (4,2), 19.11°, 4.14 Å	Nanotubes, mono-species	ASE
100.	Carbon nanotube	chiral, (4,3), 25.28°, 4.76 Å	Nanotubes, mono-species	ASE
101.	Carbon nanotube	chiral, (5,1), 8.95°, 4.36 Å	Nanotubes, mono-species	ASE
102.	Carbon nanotube	chiral, (5,2), 16.1°, 4.89 Å	Nanotubes, mono-species	ASE
103.	Carbon nanotube	chiral, (5,3), 21.79°, 5.48 Å	Nanotubes, mono-species	ASE
104.	Carbon nanotube	chiral, (6,1), 7.59°, 5.13 Å	Nanotubes, mono-species	ASE
105.	Carbon nanotube	chiral, (6,2), 13.9°, 5.65 Å	Nanotubes, mono-species	ASE
106.	Carbon nanotube	chiral, (7,1), 6.59°, 5.91 Å	Nanotubes, mono-species	ASE
107.	Carbon nanotube	zigzag, (6,0), 0.0°, 4.7 Å	Nanotubes, mono-species	ASE
108.	Carbon nanotube	zigzag, (7,0), 0.0°, 5.48 Å	Nanotubes, mono-species	ASE

Table S6. Complete list of prototypes (part III) include in the training set of this work. For the carbon nanotubes, the symmetry column specifies the configuration type (chiral, zigzag or armchair) together with the corresponding chiral numbers (n,m), the chiral angle θ and the nanotube diameter.