Title

Data-driven approach for synchrotron X-ray Laue microdiffraction scan analysis

Journal

Acta Crystallographica Section A: Foundations and advances, 75(6)

ISSN

0108-7673

Authors

Song, Yintao
Tamura, Nobumichi
Zhang, Chenbo
et al.

Publication Date

2019-11-01

DOI

10.1107/s2053273319012804

Peer reviewed

Data-driven approach for synchrotron X-ray Laue microdiffraction scan analysis

Yintao Song,[a] Nobumichi Tamura,[b] Chenbo Zhang,[c] Mostafa Karami[c] and Xian Chen [c]*

[a] *1170 Foster City Blvd., Foster City, CA United States,* [b] *Advanced Light Source, Lawrence Berkeley National Lab, Berkeley, CA, United States, and* [c] *Mechanical and Aerospace Engineering, Hong Kong University of Science and Technology, Hong Kong. E-mail: xianchen@ust.hk*

Abstract

We propose a novel data-driven approach for analyzing synchrotron Laue X-ray microdiffraction scans based on machine learning algorithms. The basic architecture and major components of the method are formulated mathematically. We demonstrate it through typical examples including polycrystalline $BaTiO_3$, multiphase transforming alloys and finely twinned martensite. The computational pipeline is implemented for beamline 12.3.2 at the Advanced Light Source, Lawrence Berkeley National Lab. The conventional analytical pathway for X-ray diffraction scans is based on a slow pattern by pattern crystal indexing process. This work provides a new way for analyzing X-ray diffraction 2D patterns, independent of the indexing process, and motivates further studies of X-ray diffraction patterns from the machine learning prospective for the development of suitable feature extraction, clustering and labeling algorithms.

1. Introduction

X-ray crystallography is a fundamental tool in modern technologies for identifying or solving the crystalline structures of solids ever since the discovery of crystal diffraction in 1912.(Friedrich *et al.*, 1912) Today's third generation synchrotron radiation facilities produce highly collimated and high-brilliance X-rays beams, that can be focused down to sub-micrometer sizes, opening the way to spatially resolved quantitative studies of materials microstructures.(Yun *et al.*, 1999; Hignette *et al.*, 2005) Scanning Laue x-ray microdiffraction using a pink or white X-ray beam is a technique that emerged in the late 1990s at synchrotron facilities. This techonology has been exploited to map the distribution of materials structural properties, such as crystal phase identity, crystal grain orientation, lattice distortion and degree of crystallinity. (Tamura *et al.*, 2003; Ulrich *et al.*, 2011) So far, this technique has been successfully implemented at the Advanced Light Source (ALS) of the Lawrence Berkeley National lab, the Advanced Photon Source (APS) of Argonne National Lab, the European Synchrotron Radiation Facility (ESRF), the Canadian Light Source (CLS) and the Taiwan Photon Source (TPS) and is in development elsewhere. This technique enables the use of micro X-ray beam as a scanning probe to quantitatively analyze both structural and topographic information of solid crystalline materials. (Tamura *et al.*, 2003; Chen *et al.*, 2016)

Within the scanned area, every point illuminated by the micro-size x-ray white beam gives rise to a diffraction pattern captured by a 2-dimensional (2D) detector with fast acquisition and short read-out time (typically a second or less per point in total). The data collected by the 2D detector is a single channel (gray-scale) image, called a *Laue pattern*. The conventional way of analyzing a Laue microdiffraction scan is to treat each pattern independently: indexing all or the majority of the reflections in each Laue pattern with the knowledge of the crystal structure (space group, atomic types and unique positions within unit cell and lattice parameters). By performing

the indexation for all the Laue patterns in a scan, the crystal orientation (or grain structure) distribution of the material can be obtained and displayed as quantified color maps.

The success of the crystallographic analysis for a Laue microdiffraction scan strongly depends on the indexing result of individual Laue patterns. For cubic structure, the indexing procedure is straightforward if the space group and Wyckoff positions (Wyckoff, 1922) are known. Due to the nature of white beam diffraction, the values quantifying the absolute size and shape of unit cell do not contribute much to the indexing of reflections diffracted by the cubic structure (for instance, two fcc crystals with comparable lattice parameters and identical orientation will give nearly identical Laue patterns). However, for crystal structures with symmetry lower than cubic, the relative sizes and angles of the unit cell (*i.e.* a/b, a/c, α, β and γ) play an important role in the indexing procedure. For these crystals, many uncertainties can arise in the indexing results when the lattice parameters are not well known and at least one of the lattice parameters is fairly large. Slight perturbations of the lattice parameters might result in different indices corresponding to the same reflection in the Laue pattern, *i.e.* misindexation. Sometimes, the indexing procedure would fail when the initial guess of the lattice parameters is far off from the true values for the tested material.

The outcome of modern X-ray microdiffraction experiments is rich in data. A typical 2D scan generates 1k to 15k diffraction patterns. From the computational point of view, the iterative indexing calculations of individual Laue patterns are often redundant and time expensive for large datasets. The analysis and indexing results of all Laue patterns within an iso-oriented spatial domain (a crystal grain) are almost identical. Using crystallographic analysis tools such as XMAS (X-ray microdiffraction analysis software) (Tamura, 2014), the iso-oriented regions with specific color labels are identified. However, when these tools are applied to label a large area comprised of

thousands of hundreds of Laue patterns, it usually takes days to finish the calculation by the optimized indexing algorithm running on a desktop computer, and synchrotron facilities have now opted to use parallel versions of the indexing code running on powerful computational GPU or CPU clusters. However, scientists who collect data at the beamline do not necessarily have access to these clusters at their home institution. If the scanned domain comprises of multiple phases with different orientations, the complexity of the analysis and the computational time will dramatically increase. For instance, the processing of the $BaTiO_3$ scan mentioned below (Figure 1) and consisting of 6000 Laue patterns take about 15 minutes using 600 nodes on a cluster, but the analysis would have taken 6 days on a regular desktop machine.

In this paper, we propose a data-driven approach to abate the burden of indexing calculation for the analysis of synchrotron Laue X-ray microdiffraction scans so that analysis can be performed quickly without the need of a supercomputer. The main goal is to outline the methodological basis of the computational pipeline with the application to the data segmentation for Laue X-ray microdiffraction experiment by machine learning algorithms. Our method has been implemented on beamline 12.3.2 at the Advanced Light Source, Lawrence Berkeley National Lab, but can be easily extended to similar X-ray diffraction experiments at any facility.

2. Methodology

Our idea originates from the nature of X-ray microdiffraction results. The spatial distribution of certain property is usually a piecewise constant function. For example, Figure 1 shows the orientation map of a martensite polycrystal $BaTiO_3$ represented by the angle between the a crystal axis and the normal to the sample surface. The regions of same color correspond to the same crystallographic orientation up to small variations. The color map is calculated and generated by XMAS (Tamura, 2014), reveal-

ing the orientation distribution, morphology of grain boundaries and microstructure within each of the grains. Given the unstrained lattice parameters and the stiffness tensor of the tested material, the orientation map can be converted into a strain or stress map.(Chen *et al.*, 2016) The piecewise scalar/vectorized color values of these maps represent certain physical property of the material. Each of the color values corresponds to a specific crystallographic indexing. Within each iso-oriented region (corresponding to an iso-indexing cluster), only one delegate Laue pattern needs to be chosen for crystallographic analysis. The indexing of the rest of the patterns in the cluster is then automatically determined.

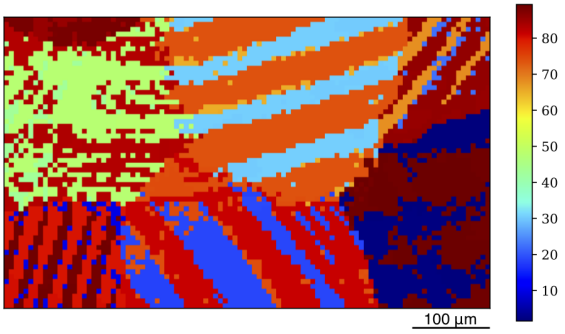

Fig. 1. Orientation map of $BaTiO_3$ from complete indexing. Scanning step size is 5μm in both directions.

In this section, we propose an indexing-free data segmentation method for X-ray microdiffraction scans. The overall process pipeline is illustrated in Figure 2. In principle, an iso-indexing cluster should consist of similar Laue patterns. From the experiment, we obtain a set of single channel images, from which a learning model can be designed to extract the features to identify their similarities. Using a machine learning algorithm, these images are classified into a set of clusters based on the extracted features. Then the inverse map of the clusters in feature space naturally forms the pre-index segmentation of the scan in spatial domain.

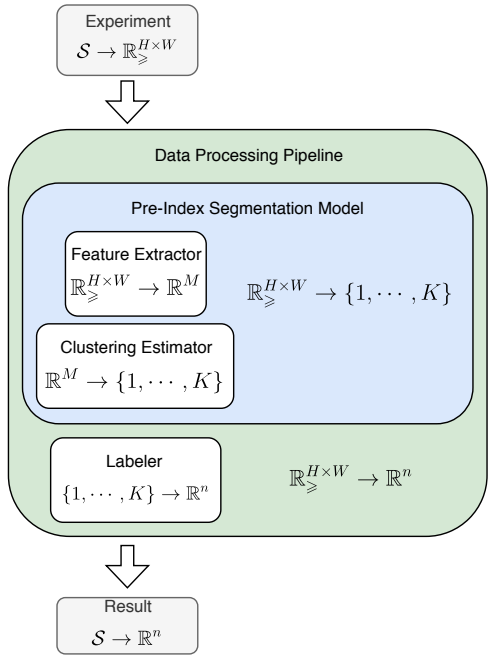

Fig. 2. Analysis pipeline of the data-driven method for X-ray Laue diffraction experiments.

In order to clearly describe the clustering and labeling methods and algorithms in our approach, we provide the following formal definitions for the X-ray Laue microdiffraction experiment.

Definition 1. The scanned area on the specimen is called the *specimen domain* $\mathcal{S}$. It is represented by a 2D mesh grid $\{1,\cdots,N_x\}\times\{1,\cdots,N_y\}\subset\mathbb{N}^2$. N_x and N_y are the number of steps along x and y direction respectively. The total number of grid points (scanned locations) is $N = N_x N_y$.

The step size along either directions is a small real number underlying the resolution of the Laue microdiffraction. Regarding the data segmentation, the values of step sizes along x and y directions do not affect the clustering result. Therefore, the specimen domain is considered as a 2D integer domain.

Definition 2. A *Laue pattern*, or simply a *pattern*, at a grid point (x, y) is a $H \times W$

gray-scale image $I_{x,y} \in \mathbb{R}_{\geqslant}^{H\times W}$. A *Laue diffraction experiment*, or simply an *experiment* is a mapping $\mathcal{I} : \mathcal{S} \to \mathbb{R}_{\geqslant}^{H\times W}$.

Definition 3. A *property map* on the specimen domain $\mathcal{S}$, is a function $f : \mathcal{S} \to \mathbb{R}^n$. n is the dimension of the property.

An experiment itself is a property map of the dimension $H \times W$. But it is not an interesting one. In practice, the goal is almost always to find a property map with some physical significance, such as the orientation of the crystal. The procedure being introduced in this paper is going to find an approximate map that is close to the true map but requires much less, if any, indexing effort.

Definition 4. A *feature extractor* V maps a diffraction pattern I to a M dimensional feature vector $\mathbf{v}$. That is

$$V : \mathbb{R}_{\geqslant}^{H\times W} \to \mathbb{R}^M. \tag{1}$$

$\mathbb{R}^M$ is the *feature space*. M is also called the *number of features*. Combined with an experiment $\mathcal{I}$, we get the *feature map* $v = V \circ \mathcal{I}$. The map

$$v : \mathcal{S} \to \mathbb{R}^M \tag{2}$$

gives the feature vector $v(x, y)$ at each grid point (x, y). The feature vectors in the image $v(\mathcal{S}) \subset \mathbb{R}^M$ are called the *feature samples*.

Usually, $M \ll H \times W$ in practice, which is the motivation of extracting features out of diffraction patterns.

Definition 5. A *feature transformation* T maps a feature vector $\mathbf{v}_1 \in \mathbb{R}^{M_1}$ to another feature vector $\mathbf{v}_2 \in \mathbb{R}^{M_2}$. *i.e.* $T : \mathbb{R}^{M_1} \to \mathbb{R}^{M_2}$.

A straightforward corollary from the above definition is that the composition $\tilde{V}$ and

$\tilde{v}$ given by

$$\begin{cases} \tilde{V} = T_n \circ \cdots \circ T_i \circ \cdots \circ T_1 \circ V \\ \tilde{v} = T_n \circ \cdots \circ T_i \circ \cdots \circ T_1 \circ v \\ V : \mathbb{R}_{\geqslant}^{H \times W} \to \mathbb{R}^{M_0} \\ v : \mathcal{S} \to \mathbb{R}^{M_0} \\ T_i : \mathbb{R}^{M_{i-1}} \to \mathbb{R}^{M_i} \end{cases} \tag{3}$$

are still a feature extractor and a feature map to M_n features.

Definition 6. A *clustering estimator*, or simply an *estimator*, g, of K clusters in the feature space $\mathbb{R}^M$ is a map

$$g : \mathbb{R}^M \to \{1, \cdots, K\}. \tag{4}$$

The integer K is the *number of clusters*. For each $k \in \{1, \cdots, K\}$, the *cluster* $\mathcal{C}_k \subset \mathbb{R}^M$ is the equivalent class

$$\mathcal{C}_k = \left\{ \mathbf{v} \in \mathbb{R}^M : g(\mathbf{v}) = k \right\}. \tag{5}$$

If the feature space is extracted by the feature map v, the specimen domain $\mathcal{S}$ is partitioned into *subdomains*

$$\mathcal{S}[v]_k = \{(x, y) \in \mathcal{S} : v(x, y) \in \mathcal{C}_k\}. \tag{6}$$

When there is no confusion about v, we ignore the parameter v, and simply write $\mathcal{S}_k$.

The motivation of partitioning feature spaces into clusters is the belief that the property of interest is (almost) constant across all feature vectors in one cluster. We call this shared property the *label* of a cluster.

Definition 7. A *labeler* ℓ with n *channels* of K clusters is the map

$$\ell : \{1, \cdots, K\} \to \mathbb{R}^n \tag{7}$$

$\ell(k)$ is the *label* of cluster $\mathcal{C}_k$.

As defined in Definition 3, n is the dimension of the shared property.

Clearly, the identity mapping $\ell_{\mathrm{N}}(k) = k$ is a labeler for any number of clusters. We call ℓ_{N} the *natural labeler*, the resulting label k is the *natural label* of $\mathcal{C}_k$.

Definition 8. A triplet (v, g, ℓ) of (feature map, estimator, labeler) is called a *data processing pipeline* or simply *pipeline*. The first two phases (v, g) is called a *pre-index segmentation model*, or simply *segmentation model* or *segmentation*.

Definition 9. For a pipeline (v, g, ℓ), the property map defined as

$$\phi[v, g, \ell] = \ell \circ g \circ v. \tag{8}$$

is called the *label map* generated by the pipeline.

The label map in Definition 9, ϕ is map from spatial domain $\mathcal{S}$ to the space $\mathbb{R}^n$. This map is the approximation of the true property map from the complete indexing. By definition, this approximate map is determined by the pipeline (v, g, ℓ). In the following sections, we are going to discuss the strategies of constructing each phase of the pipeline.

Definition 10. A *centroid assignment* u assigns a feature vector to each cluster:

$$u : \{1, \cdots, K\} \to \mathbf{u}_k. \tag{9}$$

The mapped vector $\mathbf{u}_k$ is called *centroid* of the cluster $\mathcal{C}_k$.

The exact centroid assignment depends on the clustering algorithm. Normally it is the mean position of all the points (in the training set) in a cluster. Note that the centroid of a cluster may not be a feature sample (Definition 4). For example, the centroid can be the mean of all feature vectors in a cluster. Because of this, we need to introduce the concept of *delegate* samples.

Definition 11. The *delegate point* (x_k, y_k) of the cluster $\mathcal{C}_k$ is given by a *delegate assignment*

$$w : \{1, \cdots, K\} \to \mathcal{S}. \tag{10}$$

The *delegate sample* (or the *delegate feature vector*) of $\mathcal{C}_k$ is

$$\mathbf{w}_k = v(x_k, y_k). \tag{11}$$

Usually we pick the feature sample closest or equal to the centroid:

$$(x_k, y_k) = \underset{(x,y)\in\mathcal{S}_k\cap\{(x,y):I_{x,y}\text{ can be indexed}\}}{\arg\min} \|v(x,y) - \mathbf{u}_k\| \tag{12}$$

The assignment $w(k)$ from the selected delegate sample calculated by equation (12) might not exist if none of the patterns in the cluster can be indexed. Note that both the centroid assignment u and the delegate assignment w are labelers of K clusters.

The main purposes of labeling are two-fold: 1) Reduce the feature space of dimension M to a lower dimensional space $\mathbb{R}^n$ such that its distribution over the specimen domain $\mathcal{S}$ can be well presented and visualized. 2) Since the label itself is also a feature derived from the original Laue patterns, the label map creates a spatial distribution of such a derived feature. In general, this derived feature does not have any physical meaning, but it quantifies the similarities between Laue patterns. Thus, we need to index the delegate pattern for each cluster, to associate each of the label values, and therefore to create the whole label map with certain physical meaning. However, if we use a labeler that is closely related to some physical properties, then the label map is itself a distribution with physical significance. Based on the aforementioned methodology, the feature extraction and clustering of the Laue patterns are independent of the indexing process.

3. Feature extraction and transformation

In this section and the next, we are going to walk through the procedure of constructing the data processing pipeline, using a dataset from a polycrystalline multi-phase $BaTiO_3$ sample as an example. The orientation map as a result of the complete indexing shown in Figure 1 is the reference map, by which the label map generated by our data-driven approach will be assessed.

3.1. CNN Autoencoder

Original Laue patterns in our examples are images of about 1M pixels. If we use all pixels as features, the number of dimensions is too large to compute. Thus we use a Convolutional Neural Network (CNN) autoencoder (Hinton & Salakhutdinov, 2006) to reduce it into a manageable number of dimensions, called *latent features*.

An autoencoder is a dimension reduction technique for unlabeled data that consists of an encoder and a decoder. The encoder compresses each high dimensional data into a low dimensional vector, *i.e.* latent features, then the decoder, normally with a mirrored architecture compared with the encoder, inflates the low dimensional vector back to the reconstructed data in the original dimension. The autoencoder is trained by minimizing a certain distance function between the original and reconstructed data. (Rumelhart *et al.*, 1986) A CNN autoencoder uses a pair of mirrored CNNs as the encoder and decoder.

First, we crop the original images to a square shape and then shrink them to 128×128 pixels. As seen in Figure 3, a Laue pattern is an overall black image with a few sparsely distributed high intensities regions called "peaks" or "reflections". For the purpose of segmentation, the exact peak profile is not as important as the peak positions and the symmetric distribution of these peaks in a Laue pattern. Therefore reducing image size will speed up training process without losing any key features. More importantly, the encoding of a smaller sized image is much faster, which makes the following steps more efficient after training.

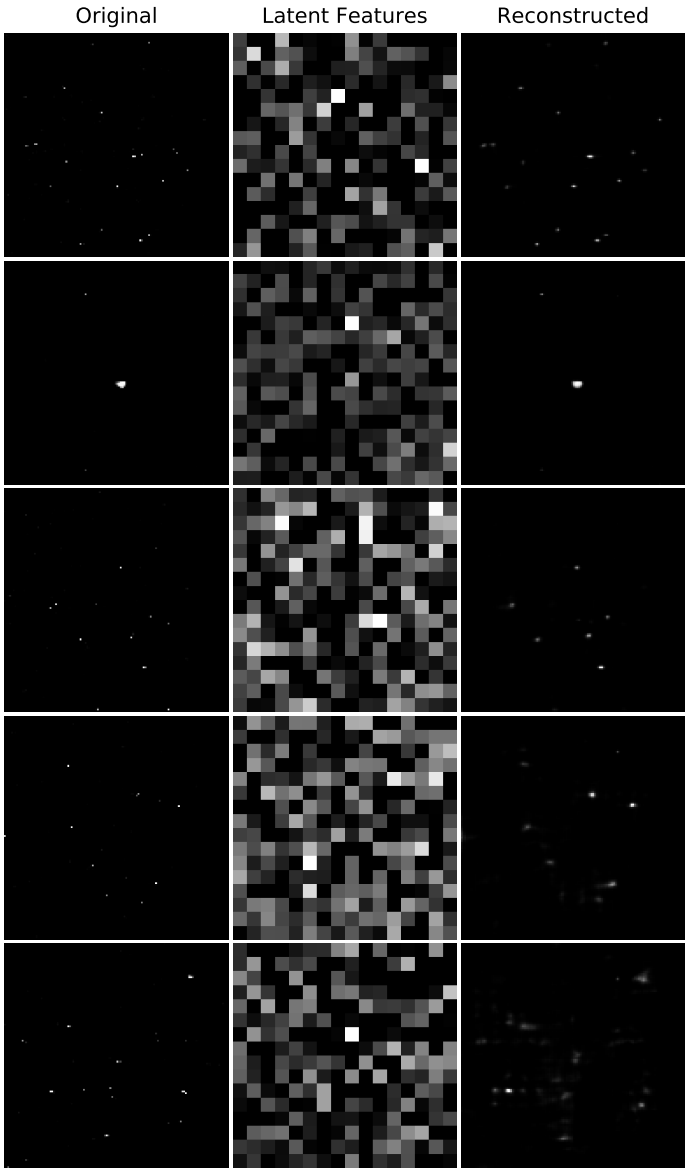

Fig. 3. Latent features extraced by autoencoder ($M_{\mathrm{AE}} = 256$).

The architecture of our CNN autoencoder is illustrated in Figure 4. Dropout layers are inserted between layers to prevent overfitting, which are omitted in the picture. The bottleneck layer of shape $1 \times 1 \times M_{\mathrm{AE}}$ represents the M_{AE} latent features.

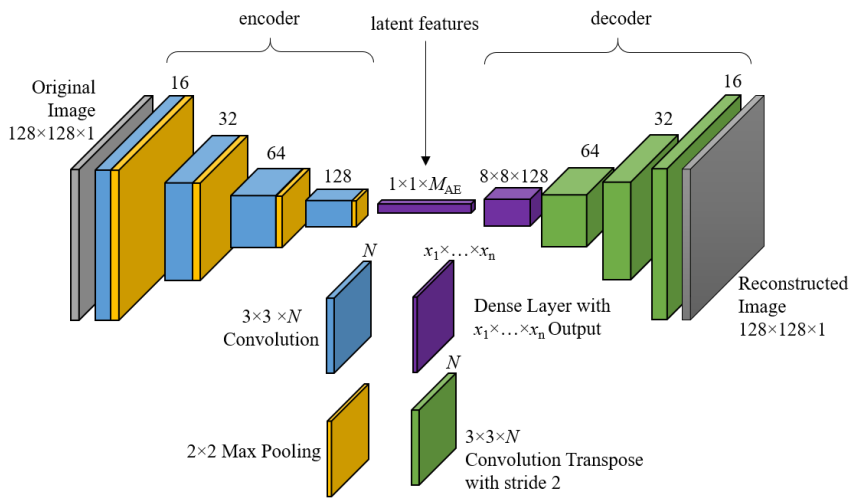

Fig. 4. Architecture of the CNN autoencoder.

We train a generic autoencoder that can distinguish the differences between any two general Laue patterns, rather than only those in one particular experiment. We use a training set of 20,000 patterns collected from over 100 different and independent experiments, and a validation set of 100 patterns that are carefully chosen to cover as many different (from human's point of view) kinds of patterns as possible. Encoding examples from one of the autoencoder we trained are shown in Figure 3. The difference between the original patterns can be captured reasonably well by the difference in latent features.

Using the notations introduced in Section 2, we have the autoencoder as a generic feature extractor

$$V_{\mathrm{AE}} : [0,1]^{128\times128} \to \mathbb{R}^{M_{\mathrm{AE}}}. \tag{13}$$

This extractor is general, which can be used for any experiments when an image preprocessing is used to convert the size and intensity to $[0,1]^{128\times128}$. Therefore, for any experiment, a feature map parameterized by M_{AE} can be generated as

$$v_{\mathrm{AE}} : \mathcal{S} \to \mathbb{R}^{M_{\mathrm{AE}}}. \tag{14}$$

3.2. PCA transformation

Principal component analysis (PCA) is an orthogonal projection of feature space to another vector space which is often of lower dimensional than the original feature space, such that the variance of the projected feature samples is maximized. (Bishop, 2006; Jolliffe, 2002) Each axis of the projected orthogonal basis is called a *principal component*. The variance of the projected feature samples along the i-th axis is called the explained variance of the i-th principal component. Without loss of generality, it is a common practice to sort the projected orthogonal basis, or the principal components, in descending order of their explained variance.

We always apply PCA transformation to the result of CNN autoencoder. So the input dimension is M_{AE}, the output dimension is $M_{\mathrm{PCA}} \leqslant M_{\mathrm{AE}}$. $M_{\mathrm{PCA}} < M_{\mathrm{AE}}$ means the feature space is truncated to the first M_{PCA} principle components. Let

$$T_{\mathrm{PCA}} : \mathbb{R}^{M_{\mathrm{AE}}} \to \mathbb{R}^{M_{\mathrm{PCA}}} \tag{15}$$

be the PCA transformation, then the composition of it and the CNN autoencoder defines the feature map

$$v_{\mathrm{PCA}} = T_{\mathrm{PCA}} \circ v_{\mathrm{AE}} : \mathcal{S} \to \mathbb{R}^{M_{\mathrm{PCA}}}. \tag{16}$$

For our study case $BaTiO_3$, we apply PCA to the latent features extracted by an autoencoder with $M_{\mathrm{AE}} = 256$, then plot the explained variance for each principal component as shown in Figure 5. Clearly, the distribution is so highly skewed that the majority of weight only concentrates in the first few components. This suggests that the truncation of the feature dimensions to the first few principal components does not affect much of the clustering accuracy for the composed feature map given in (16). This PCA truncation will be discussed in more details in the next section.

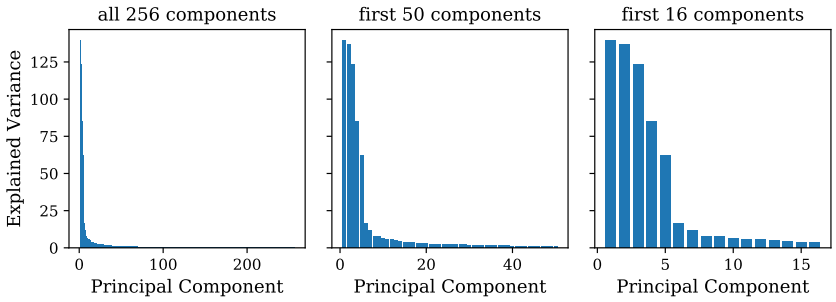

Fig. 5. Explained Variance of Principal Components in $BaTiO_3$

4. Clustering and labeling

After mapping each of the patterns into a feature vector, we can use clustering algorithms to train an estimator g. We consider two clustering techniques, namely K-Means (Bishop, 2006; Lloyd, 1982) and Bayesian Gaussian Mixture (BGM) (also known as Variational Mixture of Gaussians) (Bishop, 2006; Attias, 1999). K-Means is a fast algorithm suitable for clusters that are well separated. It is scalable to huge data sets. The number of clusters K is crucial and mandatory for a K-Means estimator. BGM on the other hand is a more advanced clustering that fits ellipsoidal clusters better than K-Means (which only assumes spherical clusters). With the usage of variational Bayesian inference, BGM is also more robust on the number of clusters. (Blei *et al.*, 2006) BGM assigns a weight to every cluster. The number of clusters with non-negligible weight (effective clusters) is smaller than the nominal number of clusters, and is stable as the nominal number of clusters increases. As a trade off, BGM is much more computationally expensive compared to K-Means. It scales poorly with the size of the data set.

According to our notation, a K-Means estimator and a BGM estimator with parameter K are denoted as g_{KM} and g_{BGM} respectively. Using either one of the two clustering algorithms combined with the feature map in (16), we obtain a pre-index segmentation

model (Definition 8). Such a segmentation has parameters $\{M_{\mathrm{AE}}, M_{\mathrm{PCA}}, K, \{KM, BGM\}\}$. For common machine learning applications, the selection of each of the parameters known as the *model selection* relies on the *metric analysis*: define and understand a metric that tells the "goodness" of a segmentation. In the following part, we are going to study model selection for supervised and unsupervised labeling.

4.1. Natural labeler

Before going into metric analysis, we inspect the raw output of the segmentation. Recall that the natural labeler of K clusters is the identity map $\ell_{\mathrm{N}}(k) = k$. We plot the label map $\phi[v, g, \ell_{\mathrm{N}}]$ for various choices of parameters, in Figure 6, 7, 8, 9. From these figures, we observe that:

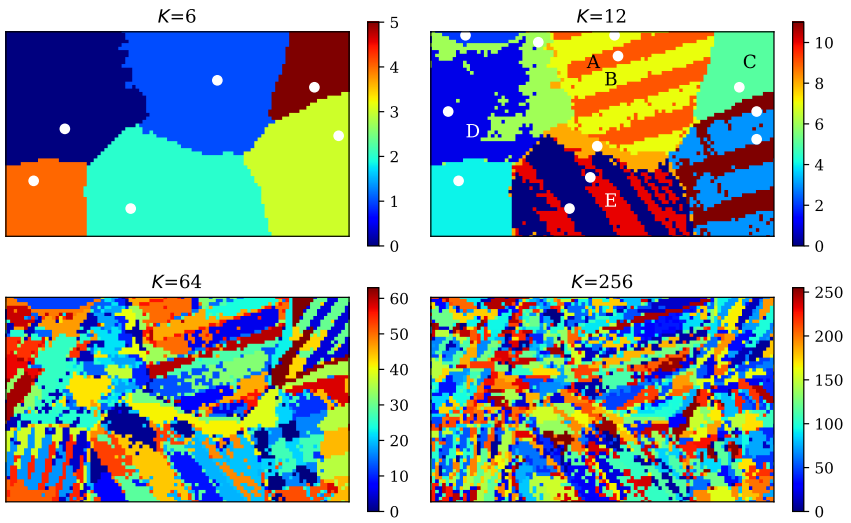

Fig. 6. K-Means clustering of $BaTiO_3$ colored by natural labeler. $M_{\mathrm{PCA}} = M_{\mathrm{AE}} = 256$. White dots are the delegate points, which is omitted in the $K = 64$ and $K = 256$ cases.

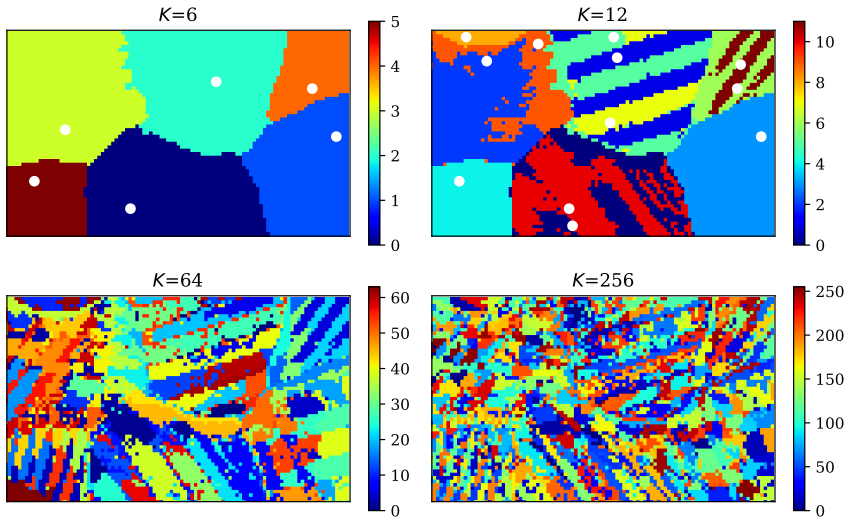

Fig. 7. Bayesian Gaussian Mixture of $BaTiO_3$ clustering colored by natural labeler. $M_{\mathrm{PCA}} = M_{\mathrm{AE}} = 256$. White dots are the delegate points, which is omitted in the $K = 64$ and $K = 256$ cases.

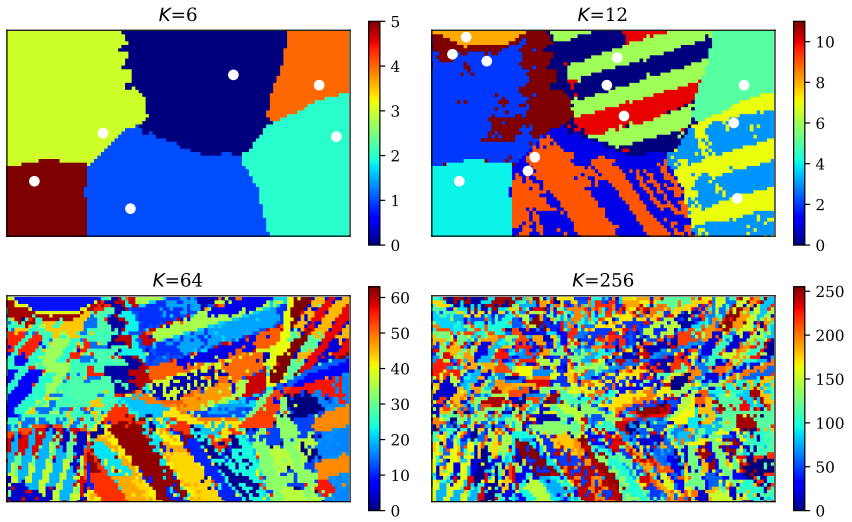

Fig. 8. K-Means clustering of $BaTiO_3$ with truncated principal components, colored by natural labeler. $M_{\mathrm{PCA}} = 32$ and $M_{\mathrm{AE}} = 256$. White dots are the delegate points, which is omitted in the $K = 64$ and $K = 256$ cases.

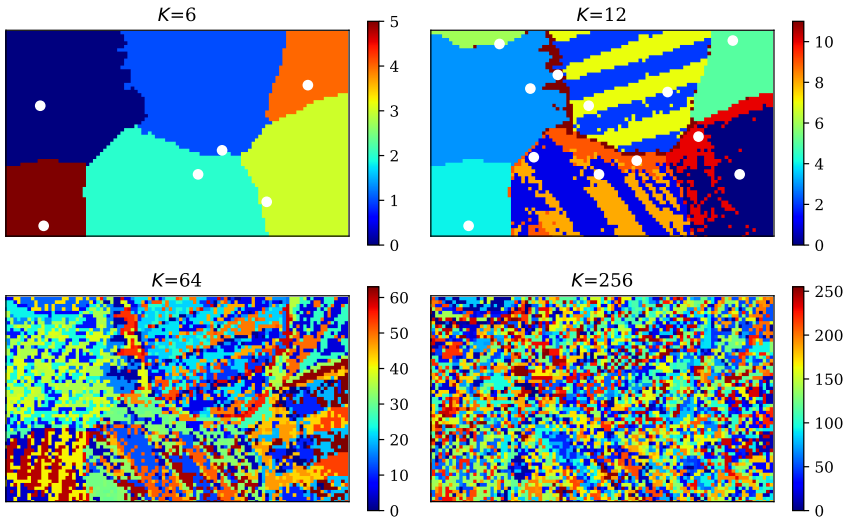

Fig. 9. K-Means clustering of $BaTiO_3$ with truncated principal components, colored by natural labeler. $M_{\mathrm{PCA}} = 8$ and $M_{\mathrm{AE}} = 256$. White dots are the delegate points, which is omitted in the $K = 64$ and $K = 256$ cases.

1. At $K = 6$, all segmentation models clearly partition the specimen domain into 6 grains.

2. At $K = 12$, all segmentation models start to reveal the twinning microstructure in some grains.

3. As K increases, more and more fine features are extracted. However when K is too large (e.g. $K = 64$ and $K = 256$), the segmentation colored by the natural labeler becomes mosaic.

4. No visible improvement or difference is observed in BGM segmentation (Figure 7) compared to K-Means segmentation (Figure 6).

5. No visible degradation or difference is observed in the K-Means segmentation using only the first 32 principal components (Figure 8) compared with K-Means using all features (Figure 6). However, K-Means corresponding to $M_{\mathrm{PCA}} = 8$ (Figure 9) is significantly noisier than that corresponding to M_{PCA}.

Given the above items 4 and 5, we will only consider K-Means with all 256 features and K-Means with the first 32 principal components in following discussions, unless otherwise mentioned.

4.2. Unsupervised labeling

When the true orientation map is not available, *e.g.* the X-ray crystallography software fails to analyze portion or all of the Laue patterns from experiments, an unsupervised labeling becomes necessary. In order to plot the segmentation result, we need to define a labeler that is independent of the true map. A careful observation of the density distribution of clusters in the feature space (Figure 10), guides us intuitively to conclude the following from the characteristics of clusters A, B, C, D and E in the $K = 12$ case in Figure 6: 1) The shape of the clusters in the feature space are close to ellipsoids, that is the density distribution of each feature is close to a Gaussian. 2) Clusters A and B as a twinning pair in the same grain, has almost the same distribution, except for the 7-th principal component, while the difference from either of them to C, D or E is significant. The second observation demonstrates that the distance between patterns in the features space has a quantitative and sensitive correlation to the physical difference between their underlying crystal structures and crystallographic orientations.

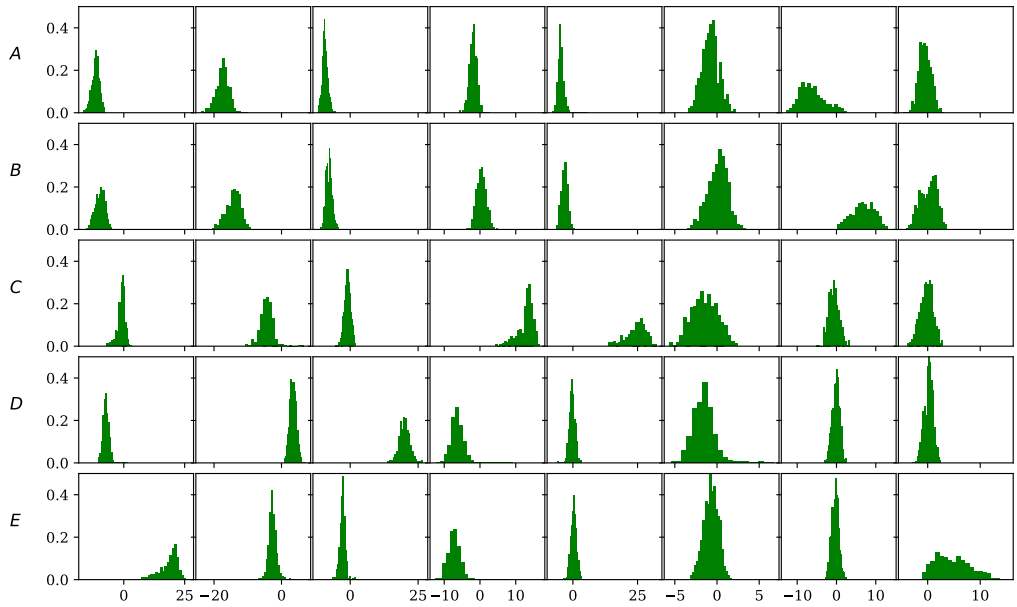

Fig. 10. Density distribution of the first 8 principal components for 5 clusters in the $K = 12$ case in Figure 6.

First, we use a PCA transformation from $\mathbb{R}^{M_{\mathrm{PCA}}}$ to $\mathbb{R}^3$ to transform the centroids of all the clusters $\{\mathbf{u}_k\}$ and truncate them to the first 3 principal components. Subsequently, we scale the range of each projected components to $[0, 255]$. The composition of such a PCA transformation and the linear scaling is defined as the *PCA labeler*, ℓ_{PCA}. ℓ_{PCA} has 3 channels. Thus, we define a coloring scheme that uses the 3 channels as the intensity of red, blue, and green channels respectively. The results corresponding to the segmentations in Figure 6 and 8 are shown in Figure 11 and 12 respectively.

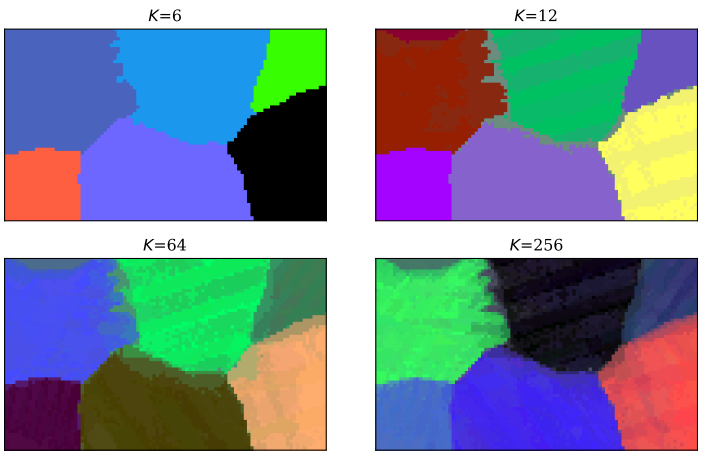

Fig. 11. K-Means clustering of $BaTiO_3$ colored by PCA labeler. $M_{\mathrm{PCA}} = M_{\mathrm{AE}} = 256$. The coloring scheme is explained in the text.

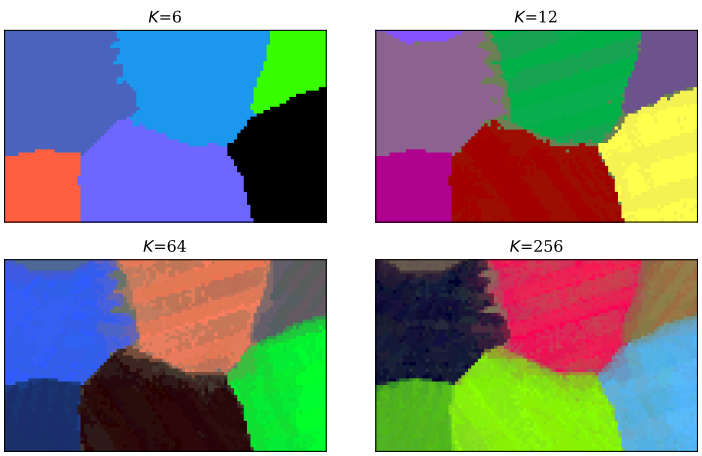

Fig. 12. K-Means clustering of $BaTiO_3$ with truncated principal components, colored by PCA labeler. $M_{\mathrm{PCA}} = 32$ and $M_{\mathrm{AE}} = 256$.

In contrast to the fuzzy color maps of large K given by the natural labeler, the maps of large K (*e.g.* $K = 64$ and $K = 256$) by the PCA labeler all converge to a constant color map, which may suggest certain intrinsic property map of the material. More importantly, the fine features in the map are stable as K increases. It suggests that without concerning the subsequent indexing effort, the more clusters used in the

segmentation, the better the pipeline extracts the spatial features in the specimen domain. Pushing this thought to an extreme, we set $K = N$. That is each sample is a cluster by itself. In other words, we directly use the first 3 principal components to color all feature samples, without any clusters. This leads to the following definition.

Definition 12. The pipeline $(v_{\mathrm{PCA}}, g_I, \ell_{\mathrm{PCA}})$ is called the *direct coloring* and the associated segmentation (v_{PCA}, g_I) is the *direct segmentation*, where the dummy estimator $g_I : v_{\mathrm{PCA}}(\mathcal{S}) \to N$ assigns each feature sample to one of the N clusters containing only this single sample.

The result of applying the direct coloring to $BaTiO_3$ is shown in Figure 13

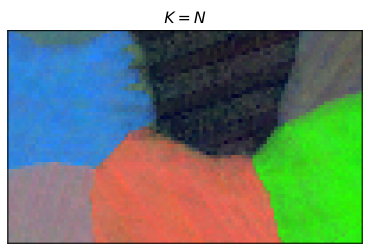

Fig. 13. Direct coloring of $BaTiO_3$ ($M_{\mathrm{AE}} = 256$).

The direct segmentation (v_{PCA}, g_I) retains the most complete information resulting from the feature map v_{PCA}. Any non-trivial clustering (v_{PCA}, g) causes certain information loss. When the conventional indexing procedure is either not available or not needed for the experiment, the direct segmentation (v_{PCA}, g_I), *i.e.* pre-clustering, with an appropriate labeler can be used to independently analyze the X-ray microdiffraction scans of any crystallographic information of the material. The PCA labeler as shown in Figure 13 is a good clustering tool in general. Even in the scenarios where we have to group feature samples into a small number of clusters, we can still use the direct segmentation as a reference to help choosing K.

In all above segmentation models, $K = 6$ consistently gives the clearest segmentation of grains. This is because the difference in the diffraction patterns across grains is much larger than that within a grain. We say this specimen has 6 *major subdomains*. In most of the materials, the major subdomains are separated by grain boundaries or phase boundaries. Figure 14 plots the Silhouette score (Rousseeuw, 1987) and the Calinski-Harabaz score (Caliński & Harabasz, 1974) for K between 2 and 20. Peaks in both scores occur near $K = 6$, which explains the aforementioned observation about major subdomains.

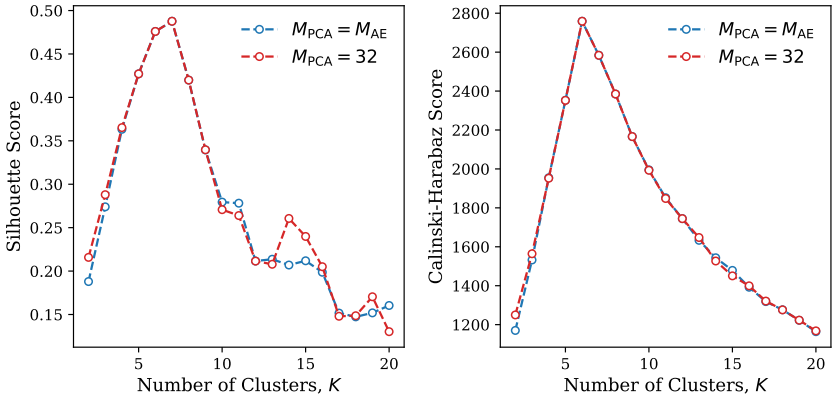

Fig. 14. Silhouette score and Calinski-Harabaz score for K-Means segmentation of $BaTiO_3$ with different K.

4.3. Supervised labeling

In a supervised labeling, we utilize the true orientation map as a result of complete indexing to assess the quality of a segmentation pipeline $\{v, g\}$. Denote the true orientation map, as shown in Figure 1, as $o(x, y)$.

Definition 13. The *indexing labeler* of a property map $f(x, y)$ gives the true value of f at the delegate point $w(k)$ for the cluster $\mathcal{C}_k$:

$$\ell_{\mathrm{I}}[f](k) = f(w(k)). \tag{17}$$

When assessing a segmentation model via supervised method, we can directly use $o(x, y)$ to evaluate $\ell_{\mathrm{I}}[o]$. In practice, to get $\ell_{\mathrm{I}}[o]$ one needs to get the true value of orientation at the delegate points $\{(x_k, y_k)\}$ by indexing only the patterns at those points. Appending the indexing labeler to a segmentation (v, g) completes a pipeline $(v, g, \ell_{\mathrm{I}}[o])$, and therefore generates a label map $\phi[v, g, \ell_{\mathrm{I}}[o]]$. We plot in Figure 15, 16 the same segmentations as in Figure 6, 8, and colored by the indexing labeler.

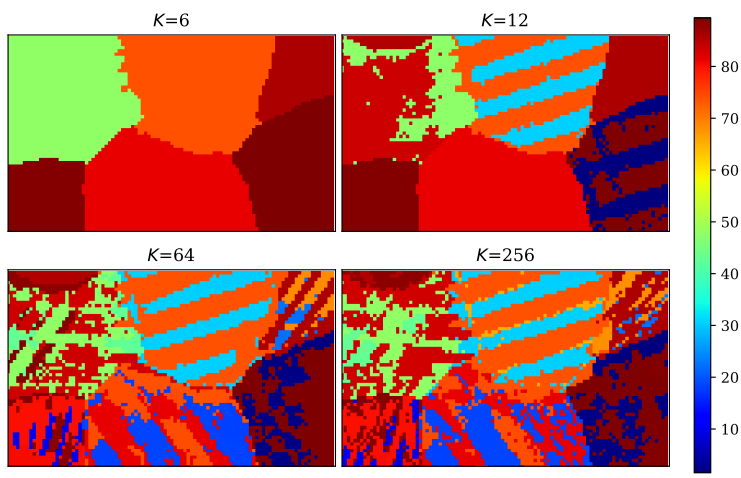

Fig. 15. K-Means clustering of $BaTiO_3$ colored by indexing labeler. $M_{\mathrm{PCA}} = M_{\mathrm{AE}} = 256$.

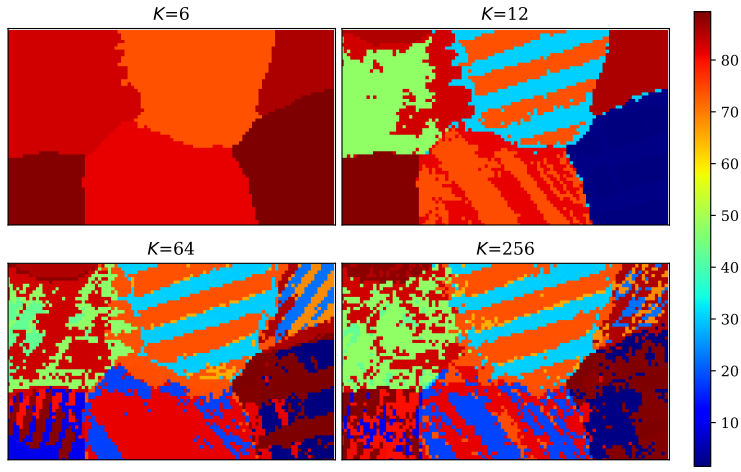

Fig. 16. K-Means clustering of $BaTiO_3$ with truncated principal components, colored by indexing labeler. $M_{\mathrm{PCA}} = 32$ and $M_{\mathrm{AE}} = 256$.

To assess the quality of the segmentation (v, g), we can check how well $\phi[v, g, \ell_{\mathrm{I}}[o]]$ approximate the true map $o(x, y)$. A natural metric for the latter check is the mean squared error between the two property maps . A less accurate but computationally cheaper test is the Kolmogorov-Smirnov distance, computed by the two-sample Kolmogorov-Smirnov test, between the two data sets $o(\mathcal{S})$ and $\phi[v, g, \ell_{\mathrm{I}}[o]](\mathcal{S})$. Figure 17 shows the mean squared error and the Kolmogorov-Smirnov distance for a wide range of K. While the mean squared error in general decreases as K increases, the reduction is very slow when K is large, *e.g.* when $K > 100$. This behavior is more prominent when it is studied by the Kolmogorov-Smirnov distance. This suggests that one can get an approximation with almost the same quality even using a much smaller number of clusters, *i.e.* much less number of patterns to index. Indeed, using just 64 or 256 delegate patterns, we get an orientation map that is fairly close to the true map resulting from indexing 6000 patterns. Nevertheless, the exact tolerant of approximation error depends on the specific physical problem of study. If the tolerance turns out to be small, we need to use a large K to achieve a higher accuracy.

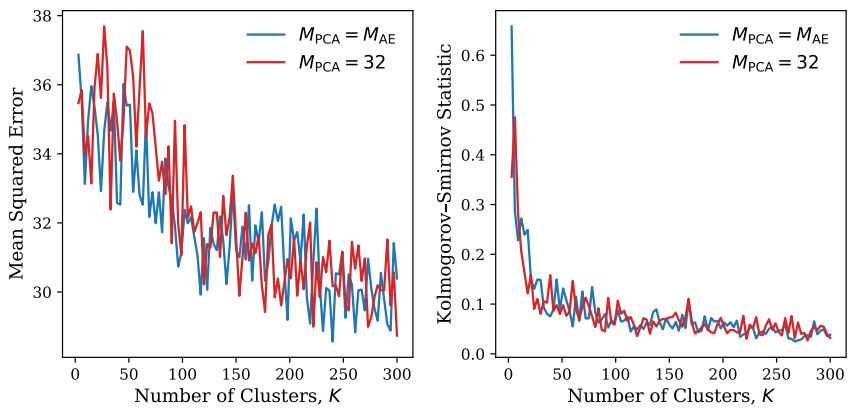

Fig. 17. Mean squared error and Kolmogorov-Smirnov distance between the clustered approximation by K-Means and the true orientation map.

4.4. Pre-Index Segmentation Procedure

Before heading to more examples, we summarize the common procedure of pre-index segmentation

1. Pick a feature extractor. Normally a CNN autoencoder with M_{AE} latent features. As we are going to see in the examples, in most cases, $M_{\mathrm{AE}} = 256$ is sufficient. In certain experiments, we might need a more complex extractor.

2. Pick a PCA transformation with $M_{\mathrm{PCA}} = M_{\mathrm{AE}}$. This completes the feature map v_{PCA}.

3. Plot direct coloring for v_{PCA}. The direct coloring gives us the first overview of the spatial features in the specimen. In some cases, the direct coloring itself has been already helpful in advancing the scientific study. The first 3 steps are quite standard for all Laue scan experiments. In general, we should decide the next steps based on the direct coloring and the particular physical problem to be solved. The following are the common options.

4. Plot Silhouette score and Calinski-Harabaz score for several K values. Try to

identify the major subdomains with the scores. Recall that major subdomains are usually separated by the grain boundaries and phase boundaries, which are the first-order microstructural features that are crucial in most experiments.

5. Visually identify some potentially important microstructural features from the direct labeling. Manipulating K to retrieve such features in label map using a reasonably small K value.

6. If the required K turns out to be big, and therefore the clustering algorithm is too slow with all features, truncate the principal components under the guidance of the distribution of explained variance of principal components.

7. Finally if needed and feasible, index the K delegate patterns. Then use the indexing labeler to get the approximated property map.

8. There is a useful strategy, which is not illustrated in the examples in later section: one can perform a coarse scan on a large area first, and get the direct coloring. By looking at the direct labeling, one may be able to locate the area of interest where a dense scan in a small area can be performed.

5. Examples

In this section we investigate 3 additional examples using the pre-index segmentation algorithm to analyze the Laue microdiffraction data collected at the Advanced Light Source Beamline 12.3.2, Lawrence Berkeley National Lab. Experimental details on the data collection procedure and description of the beamline apparatus have been published elsewhere (Tamura *et al.*, 2003; Tamura, 2014). Unless mentioned otherwise, we will use the feature map of $M_{\mathrm{PCA}} = M_{\mathrm{AE}} = 256$.

5.1. CuAlMn alloy

This alloy with atomic composition 72.5% Cu – 17.2% Al – 10.3% Mn undergoes a martensitic phase transformation at -7°C. We conducted the synchrotron microdiffraction scan at room temperature on an area consisting of several austenite grains. The austenite crystal structure can be well indexed by symmetry $Fm\bar{3}m$, and we use the XMAS parallel analysis algorithm to index the whole data set and generate the true orientation map in Figure 18a. This example shows that the pre-index segmentation algorithm correctly captures the microstructure of a single phase, multi-grains material regardless of the types of labeler and number of clusters. In the true orientation map, there exist some points, especially those near the grain boundaries where only a portion (70%) of diffraction peaks were indexed by the crystallographic analysis. By pattern segmentation, these points can be identified, and the grain boundaries are better resolved. In addition, the scanned area consists of three main clusters based on the Silhouette and Calinski-Harabaz score (Figure 19). By increasing the number of clusters, the detailed microstructures in each of the main grains can be resolved better, e.g. Figure 18b, d and f.

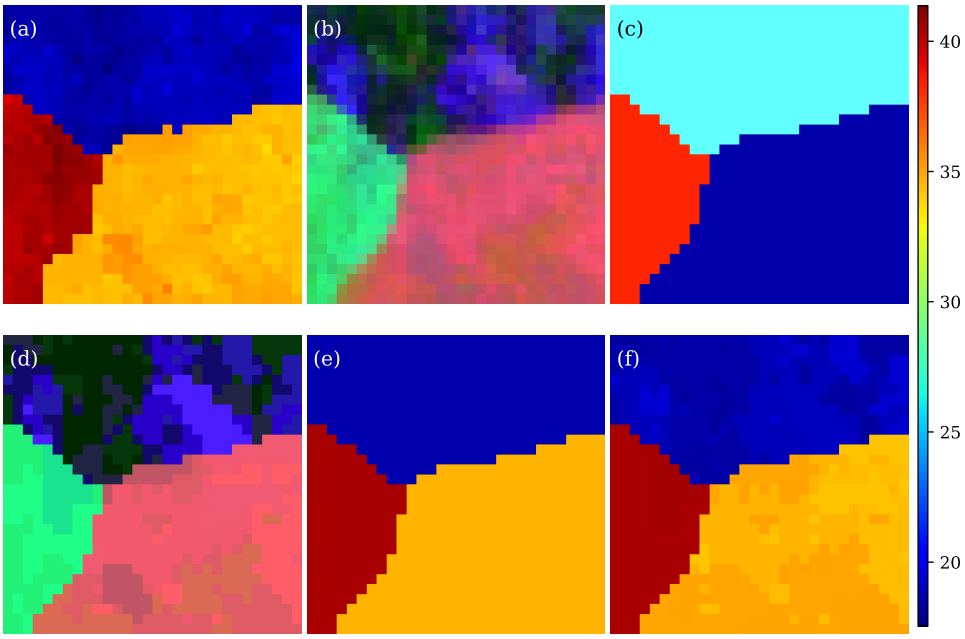

Fig. 18. In CuAlMn specimen: (a) True map. (b) Direct coloring. (c) PCA labeler with $K = 3$. (d) PCA labeler with $K = 16$. (e) Indexing labeler with $K = 3$. (f) Indexing labeler with $K = 16$. The color bar on the right is for (a)(e)(f).

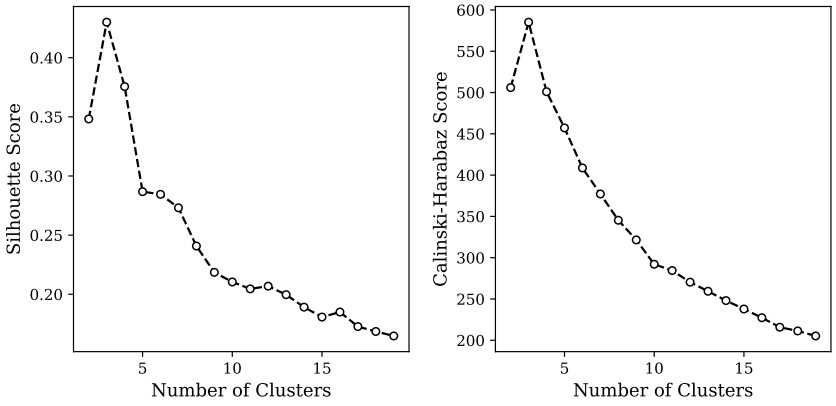

Fig. 19. Silhouette score and Calinski-Harabaz score for C5 with different K

5.2. AuCuZn alloy

The phase-transforming alloy $Au_{29}Cu_{26}Zn_{45}$ is an example having the complex interface morphology between austenite and martensite phases. At around -40°C, the cubic austenite transforms to the monoclinic martensite. To generate the true orientation map, we have to run the indexing algorithm twice for the same scanned area: one for cubic symmetry and the other for monoclinic symmetry corresponding to the Figure 20a and b respectively. By checking Silhouette score and Calinski-Harabaz score (Fig.21), there are only 2 major subdomains. A PCA labeling of $K = 2$ clustering clearly reveals the phase boundary (Figure 20d) But the corresponding indexing labeler fails to resolve the boundary clearly (Figure 20g) because the delegate selected for the whole martenstite cluster is failed to be indexed and can not represent the martensite region consisting of the martensite variants with different orientations. When we increase K to 10 and 64, both the PCA labeler (Figure 20e, f) and the indexing labeler (Figure 20h, i) reveals finer and richer microstructures of martensite. The indexing labeler at large number of clusters starts converging to the true map.

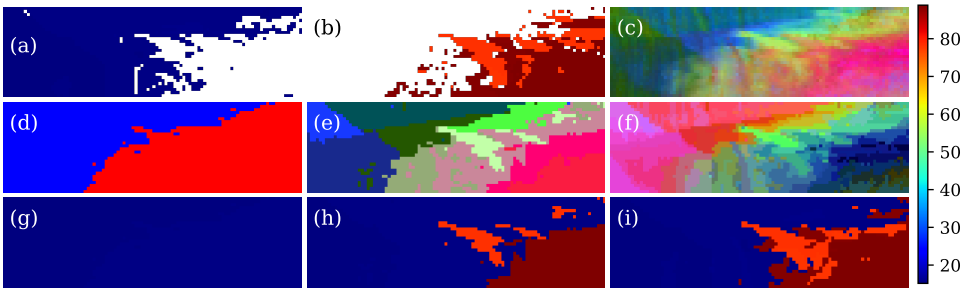

Fig. 20. In $Au_{29}Cu_{26}Zn_{45}$ alloy: (a) True map for austenite. (b) True map for martensite. (c) Direct coloring. (d)(e)(f) PCA labeler with $K = 2$, 10 and 64. (g)(h)(i) Indexing labeler with $K = 2$, 10 and 64. The color bar on the right is for (a)(b)(g)(h)(i).

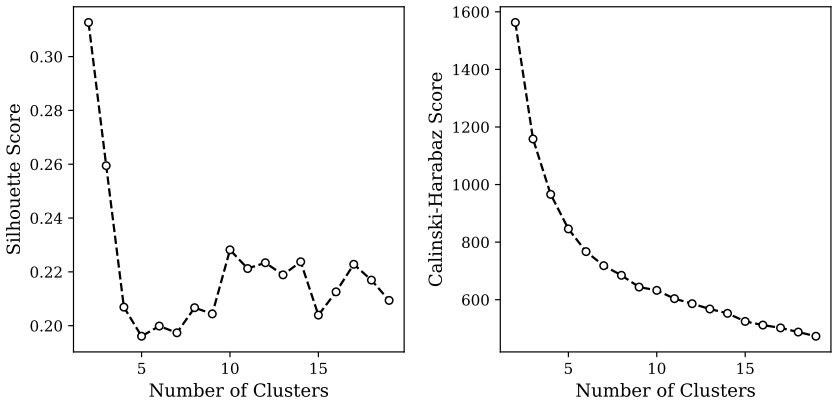

Fig. 21. Silhouette score and Calinski-Harabaz score for $Au_{29}Cu_{26}Zn_{45}$ alloy with different K

5.3. CuAlNi alloy

CuAlNi alloy is in martensite phase made up of fine twins. Since the fineness of some of the twinning structure is beyond the resolution of the X-ray microdiffraction, the indexing was not very successful. As shown in Figure 22a, There is a large area failed to be indexed. One can only roughly see certain vertical features that might correspond to the twins. In this case, the direct coloring is strongly preferred because of the lack of true map. The segmentation result by the direct coloring is shown in Figure 22b, which clearly reveal the twin feature of the specimen. The Silhouette score and Calinski-Harabaz score (Figure 23) suggests 2 major subdomains. Figure 22c is $K = 2$ clustering colored by the PCA labeler, which shows clear twin boundaries. In contrast, because of poor indexing, the indexing labeling (Figure 22f) is not applicable in this case. As we increase K to 12 and 64, the PCA labeler (Figure 22d, e) starts to show finer and finer features of twinning microstructures.

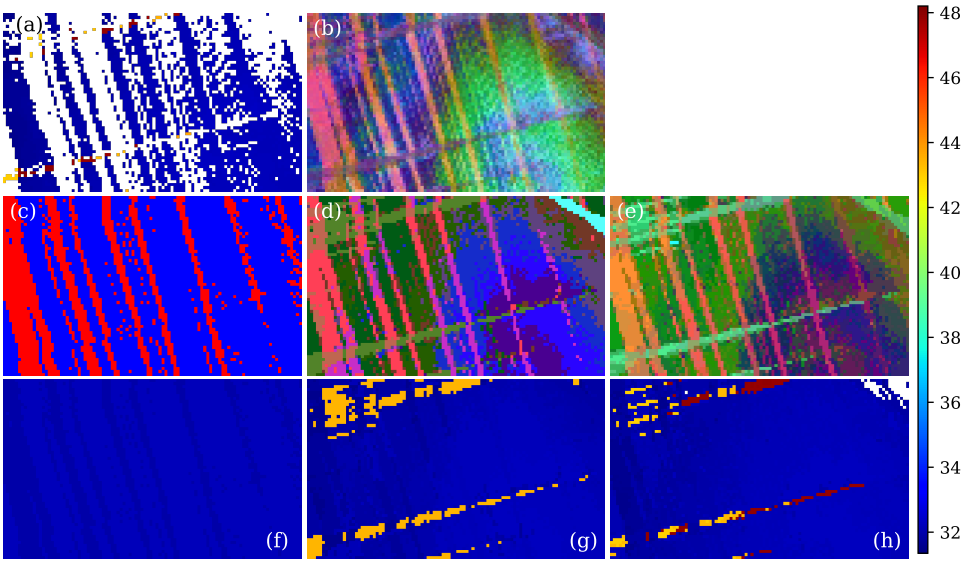

Fig. 22. In CuAlNi alloy: (a) True map. (b) Direct coloring. (c)(d)(e) PCA labeler with $K = 2$, 12 and 64. (f)(g)(h) Indexing labeler with $K = 2$, 12 and 64. The color bar on the right is for (a)(f)(g)(h).

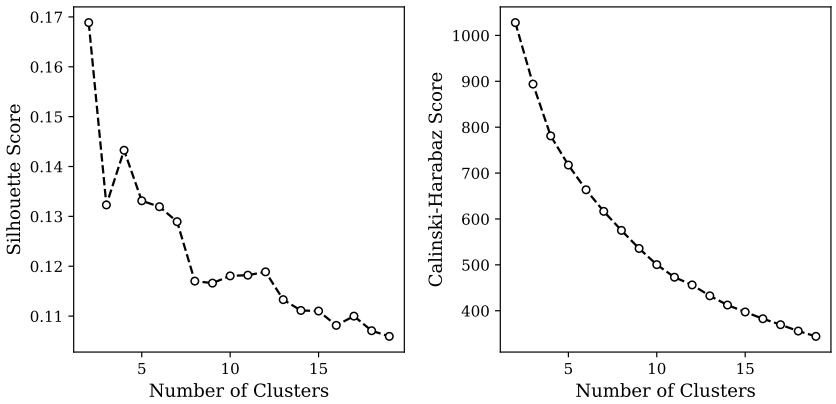

Fig. 23. Silhouette score and Calinski-Harabaz scores for CuAlNi with different K

6. Real-time Performance

As stated earlier, one of the motivations of the pre-index segmentation is to enable real-time analysis of Laue XRD data. To justify the feasibility, we list performance

profile in Table 1. In a real application, feature extraction can be done on a streaming fashion: extract features from a pattern right after it is taken by the detector. Then direct coloring and some other label maps can be computed immediately. If further tuning is needed, interactive re-clustering is possible, given the speed of computation.

Table 1. *Performance of each phase in a pre-index segmentation pipeline.*[1]

Study Case	Data Size	Feature Extraction[2,3]	Clustering[2,4]
CuAlMn	30×30	45	2.3 (3); 5.6 (16)
AuCuAn	30×100	63	2.7 (2); 6.7 (10); 14.3 (64)
CuAlNi	60×100	104	3.5 (2); 12.7 (12); 23.4 (64)

[1] Measured in a desktop environment with Intel i7-8700K (6 physical cores at 3.7GHz) and 16GB RAM.
[2] In the unit of second.
[3] Using 12 parallel jobs.
[4] The numbers in the parenthesis are K values.

7. Conclusion

In this paper, we proposed a machine learning based data processing pipeline for synchrotron Laue X-ray microdiffraction experiments. We formalized the pipeline to consist of 3 phases: feature extraction, clustering, and labeling. We then demonstrated the procedure of getting an approximate property map from Laue patterns in 4 different examples with distinct types of materials. The results are promising and the performances are impressive. A real-time data processing platform could be built on top of this pipeline.

Each of the phases can be further studied separately. For feature extraction, other CNN architectures can be explored. One could also try other feature extraction methods with more direct physical meaning, such as using the complete list of peak positions, intensities and shapes. For clustering, one could study other clustering algorithms, such as Gaussian Mixtures, DBSCAN, Mean Shift, etc. Also, advanced image segmentation techniques can be utilized. For example, Markov Random Field is the state of the art statistical model for image segmentation. For labeling, one of the lesson learned from this paper is that a good labeler mapping the feature space into a low

dimensional space with dimension less than or equal to 3 greatly helps us to visualize the property map, even without indexing. Finding better labelers, which includes finding the relationship between hidden feature space and the true physical properties, should be a persistent goal for future research.

Acknowledgements C. Z, M. K. and X. C. thank the financial support of the HK Research Grant Council under GRF Grant 16207017 and 26200316. Beamline 12.3.2 and the Advanced Light Source was supported by the Office of Science, Office of Basic Energy Sciences, of the U.S. Department of Energy under Contract no. DE-AC02-05CH11231.

References

Attias, H. (1999). In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pp. 21–30. Morgan Kaufmann.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Blei, D. M., Jordan, M. I. *et al.* (2006). *Bayesian analysis*, **1**(1), 121–143.

Caliński, T. & Harabasz, J. (1974). *Communications in Statistics-Theory and Methods*, **3**(1), 1–27.

Chen, X., Dejoie, C., Jiang, T., Ku, C.-S. & Tamura, N. (2016). *MRS bulletin*, **41**(6), 445–453.

Friedrich, W., Knipping, P. & Laue, M. (1912). *In Sitzungsber. Kgl. Bayer: Akad. Wiss. pp. 303-322*, pp. 303–322.

Hignette, O., Cloetens, P., Rostaing, G., Bernard, P. & Morawe, C. (2005). *Review of scientific instruments*, **76**(6), 063709.

Hinton, G. E. & Salakhutdinov, R. R. (2006). *Science*, **313**(5786), 504–507.

Jolliffe, I. M. (2002). *Principal Component Analysis*. Springer, 2nd ed.

Lloyd, S. (1982). *IEEE Transactions on Information Theory*, **28**(2), 129–137.

Rousseeuw, P. J. (1987). *Journal of Computational and Applied Mathematics*, **20**, 53–65.

Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1986). *Nature*, **323**(6088), 533.

Tamura, N. (2014). In *Strain and dislocation gradients from diffraction: Spatially-Resolved Local Structure and Defects*, pp. 125–155. World Scientific.

Tamura, N., MacDowell, A., Spolenak, R., Valek, B., Bravman, J., Brown, W., Celestre, R., Padmore, H., Batterman, B. & Patel, J. (2003). *Journal of synchrotron radiation*, **10**(2), 137–143.

Ulrich, O., Biquard, X., Bleuet, P., Geaymond, O., Gergaud, P., Micha, J., Robach, O. & Rieutord, F. (2011). *Review of scientific instruments*, **82**(3), 033908.

Wyckoff, R. W. G. (1922). *The Analytical Expression of the Results of the Theory of Space-groups*. No. 318. Carnegie institution of Washington.

Yun, W., Lai, B., Cai, Z., Maser, J., Legnini, D., Gluskin, E., Chen, Z., Krasnoperova, A. A., Vladimirsky, Y., Cerrina, F. & et al. (1999). *Review of Scientific Instruments*, **70**(5), 2238–2241.

Synopsis

1

A novel data-driven approach for synchrotron Laue X-ray microdiffraction scans based on machine learning algorithms.