

ARTICLE OPEN



Parametric simulation of electron backscatter diffraction patterns through generative models

Zihao Ding¹ and Marc De Graef¹✉

Recently, discriminative machine learning models have been widely used to predict various attributes from Electron Backscatter Diffraction (EBSD) patterns. However, there has never been any generative model developed for EBSD pattern simulation. On one hand, the training of generative models is much harder than that of discriminative ones; On the other hand, numerous variables affecting EBSD pattern formation make the input space high-dimensional and its relationship with the distribution of backscattered electrons complicated. In this study, we propose a framework (EBSD-CVAE/GAN) with great flexibility and scalability to realize parametric simulation of EBSD patterns. Compared with the frequently used forward model, EBSD-CVAE/GAN can take variables more than just orientation and generate corresponding EBSD patterns in a single run. The accuracy and quality of generated patterns are systematically evaluated. The model does not only summarize a distribution of backscattered electrons at a higher level, but also mitigates data scarcity in this field.

npj Computational Materials (2023)9:199; <https://doi.org/10.1038/s41524-023-01143-z>

INTRODUCTION

Since its first-ever observation in 1928, electron backscatter diffraction (EBSD) has gradually become one of the most powerful tools to study crystal texture and material anisotropy¹. By analyzing the diffraction pattern after the electrons have interacted with the sample, the orientation of the material being characterized can be determined, since it is closely correlated with the trajectories of backscattered electrons, and thus their distribution in space². The modern popularity of this technique stems from the development of CCD/CMOS imaging sensors and fully automated computer-based data analytics^{3,4}; the implementation of a Hough-transform-based method⁵ replaced the tedious manual indexing of EBSD patterns. In addition to orientation mapping, another important application of EBSD is evaluating plastic deformation processes, since local lattice misorientations are closely correlated to plastic strain or strain-induced changes^{6,7}. High angular resolution EBSD (HR-EBSD) is currently the leading approach to measurement of residual strain states at a high resolution and sensitivity⁸.

Although being widely used, characterization of a material's texture via EBSD does pose some challenges. On the one hand, the requirement of a flat surface to maximize the backscatter yield makes sample preparation time-consuming and leads to a danger of introducing extra deformation in the near-surface region⁹. On the other hand, the field of view studied in EBSD is typically limited to a few square millimeters (and, often, significantly less than that) which makes it difficult to collect sufficient data to fully characterize a material's texture. Individual EBSD patterns are affected by the crystal symmetry, the 3D orientation of the crystal lattice with respect to the incident electron beam, the energy of the incident beam as well as geometrical parameters of the detector system. Thus, most applications of EBSD are discriminative tasks, i.e., to extract attributes from experimental observations through classification or regression. Statistically, the indexing algorithms aim to optimize and compute the conditional probability, $p(\text{attr}|\text{obs})$, of the attributes, given the observations. In order to extended the reach of EBSD to more data intensive

tasks, as well as studying the relationship between all the variables mentioned above and the spatial distribution of backscattered electrons, an accurate and efficient method to simulate EBSD patterns is needed.

Currently, the mainstream approach for EBSD pattern simulation is through a physics-based forward model^{10,11}; the model in¹¹ first computes the backscattered yield over all directions via a Monte Carlo simulation, and then solves the dynamical electron scattering problem for a sampling of orientations on the Kikuchi sphere; the resulting intensity distribution is referred to as the "EBSD master pattern." Individual patterns are then obtained through a gnomonic projection from the Kikuchi sphere. The algorithm is available as a core component in the open source software *EMsoft*¹² and lays a solid foundation for Dictionary Indexing (DI)^{13,14} and Spherical Indexing (SI)^{15,16}. Both indexing approaches outperform the Hough-transform-based method in terms of accuracy and robustness against noise¹⁷.

Statistically, an ideal generative method should accurately estimate $p(\text{obs}, \text{attr})$ first and then determine the conditional distribution $p(\text{obs}|\text{attr})$ via Bayes rule so that, given an arbitrary set of attributes, it can generate the corresponding observation. Since the forward model separates attributes involving the electron interactions with the sample from attributes related to the gnomonic projection from the Kikuchi sphere onto the detector, each step is only able to provide the conditional distribution $p(\text{obs}, \text{attr}_2|\text{attr}_1)$, where attr_1 and attr_2 are attributes used in two stages respectively. Thus, each EBSD master pattern is specific to the given set of attributes; in particular, master patterns are typically computed for a discrete number of microscope accelerating voltages. This feature leads to a significant increase of time consumption when attributes in attr_1 are not fixed, as the forward model needs to run simulations from scratch for each different configuration.

The past decade has witnessed a sustained rapid development of machine learning theory. As model architectures and training algorithms become mature and modularized, it is exciting to see more and more applications in science and engineering fields,

¹Department of Materials Science and Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA. ✉email: degraef@cmu.edu

including those in material characterization. Specifically in EBSD, we have proposed two models, EBSD-CNN¹⁸ and EBSDI-CNN¹⁹, to realize end-to-end and hybrid pattern indexing, respectively. Other groups have put forth models with various output spaces to predict other attributes from EBSD patterns, such as crystal symmetry²⁰ and phase identification²¹. The success of these approaches engenders confidence in the capability of deep neural networks to extract features and determine fitting functions with high-dimensional input/output spaces.

Similar to the applications implemented with non-ML methods mentioned above, almost all these ML models are still for discriminative purposes only. As they just need to distinguish decision boundaries and optimize $p(\text{attr}|\text{obs})$ directly, even if the model is not sufficiently expressive, after significant training, these discriminative approaches still lead to superior recognition performance. Different from discriminative models, which compress the information volume during the forward propagation, a generative model is intended to reproduce the observations from random noise and given attributes in a top-down approach that gradually accumulates the information²². Thus, constructing and training generative models is usually much harder. Our main purpose in this contribution is to generate EBSD patterns with particular specified attributes. Machine learning studies with a similar goal include generative models which are conditioned on class labels²³ and text²⁴, allow editing of facial features²⁵ and outdoor scenery attributes²⁶, and even support 3D-aware scene manipulation²⁷. In materials science, Ziatdinov and Kalinin^{28,29} have also shown that features related to material properties can be disentangled from characterization data via ML approaches.

In this study, we propose a deep generative model (EBSD-CVAE/GAN) to realize analytic and parametric EBSD pattern simulation. Its great flexibility and scalability in architecture makes it possible to extend the dimension of manipulated attributes under the same training algorithm. Compared with the EBSD forward model, EBSD-CVAE/GAN allows users to change more attributes involved with the formation of EBSD patterns in a single run. The model represents the progress towards the ideal generative method in statistics, i.e., narrowing down attr_1 on which the probability is conditioned. Such an approach does not only provide a way to summarize a distribution of backscattered electrons at a higher level, but also expands applications of EBSD to situations where multiple parameters are subjected to change.

RESULTS AND DISCUSSION

EBSD-CVAE/GAN architecture

Among all generative models, the variational autoencoder (VAE)³⁰ and generative adversarial networks (GAN)³¹ are most frequently used to learn the true distribution through two different divergence measurements. The manipulation of attributes is achieved by decoding the latent representation from the encoder, conditioned on the expected attributes. The key here is to disentangle the attributes with physical meanings that we want to control from other latent representations. Accordingly, the distribution presented becomes conditioned on these manipulated attributes. Conditional generative models based on VAE and GAN are collectively referred to as variants of CVAE³² and CGAN³³. Usually, the attributes manipulated are formed as a vector, which is defined as the difference between the mean latent representations with and without them. Then, by integrating the vector to a latent representation, the decoded image from the modified representation is expected to have the corresponding attributes. A potential problem is that an attribute vector may contain highly correlated attributes, inevitably leading to unexpected changes of other attributes left in the latent representation, especially for attributes in EBSD pattern formation, which could be highly complex and closely dependent. Meanwhile, it should be realized

that another difficulty in simulating EBSD patterns through generative models is the high demand of location accuracy of features, which is also encoded in the patterns and orientation.

Since the encoder-decoder architecture is viable in both CVAE and CGAN, and the decoder/generator parts in the two models share many similarities, we propose a general learning framework, combining a CVAE and a CGAN with good scalability on attributes. Figure 1 shows a schematic of its architecture. In the full model, the decoder of CVAE is shared as the generator of CGAN. Thus by adjusting the coefficient of each item in the loss function during the training, the model can readily be turned into a single CVAE, or CGAN, or a combination of both. In addition, to cope with the conditional distribution via manipulation of certain attributes and then concatenating them to the random latent representation after preliminary transformation in the generator, alongside the discriminator we also place a classifier to predict all the manipulated attributes encoded in the patterns. After the model is properly trained, only the decoder/generator part is necessary when simulating EBSD patterns, which further lowers the demands on computational resources and memory when deployed for inference.

Training of model conditioned on orientations only

From the training history in Fig. 2a, after fluctuations of Kullback–Leibler divergence (KL divergence) in the initial epochs, it can be seen that both the reconstruction loss and the KL divergence are gradually decreasing, and finally converge to a stable state. To visually represent this process, patterns generated with random orientations in different stages of training are recorded as an animation in Supplementary Movie 1. Here we choose to use random orientations in each frame, as it is the only manipulated attribute at the first stage of the study. The filters are first trained to generate band features, and then gradually the main Kikuchi bands and zone axes are formed. Finally more attention is paid to other features that are less obvious in the background. The generated patterns after over 45 epochs of training are shown in Fig. 3a. Compared with patterns generated by models trained with fixed coefficients in the loss function, even patterns generated from a standard normal distribution are highly consistent with original patterns. The overall pattern quality indicates the encoder's ability to standardize latent representation ($(\mu, \sigma) \rightarrow (0, 1)$) and the decoder's ability to recover EBSD patterns conditioned on orientation.

Generated pattern quality analysis

The quality of the generated patterns can be evaluated further with more quantitative metrics. Comparing the real and generated patterns in Fig. 3b, it can be seen in the difference part that pixels with a larger deviation are more concentrated around the zone axes and edges of the Kikuchi bands, where the contrast rapidly changes. In real patterns, the pixel value is calculated using interpolation based on the master pattern simulated with the forward model, thus the edges of the main features are usually very sharp. As mentioned in the Supplementary Information, CVAE models tend to generate blurry edges, thus the pixel deviations of these areas are larger than elsewhere. This is also in concord with Fig. 3c, the distribution of pixel values in the original and generated patterns with orientations from the testing data set. Because the original patterns have gone through a contrast limited adaptive histogram equalization (CLAHE)³⁴, the intensity distributions of the original patterns are uniform in a normalized range from 0 to 1. The patterns generated by the trained CVAE model, while looking very similar to the original ones, have a distribution with more pixels aggregating at the middle part of the intensity range. This can be explained by the choice of binary cross-entropy for the reconstruction loss, of which the optimal value is depicted by the red spline in the figure. Compared with

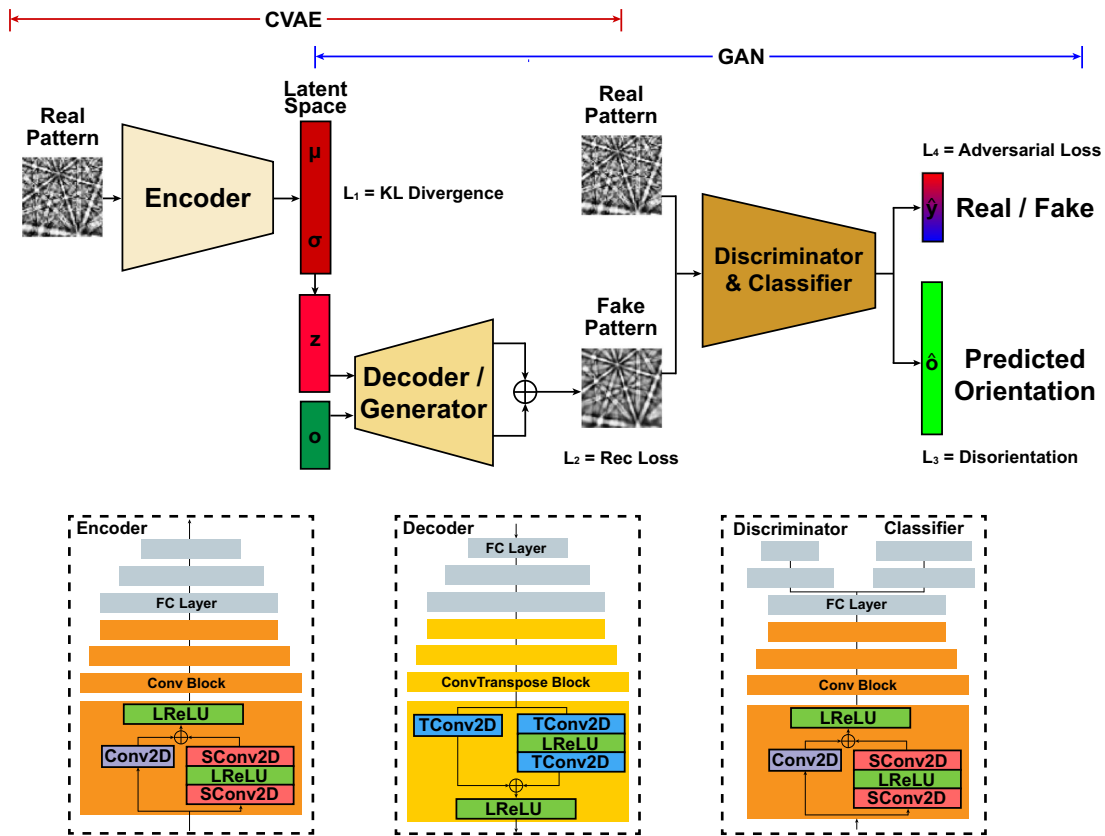


Fig. 1 Generative model schematic. Schematic of the generative model combining CVAE and GAN with attributes manipulation in the latent space.

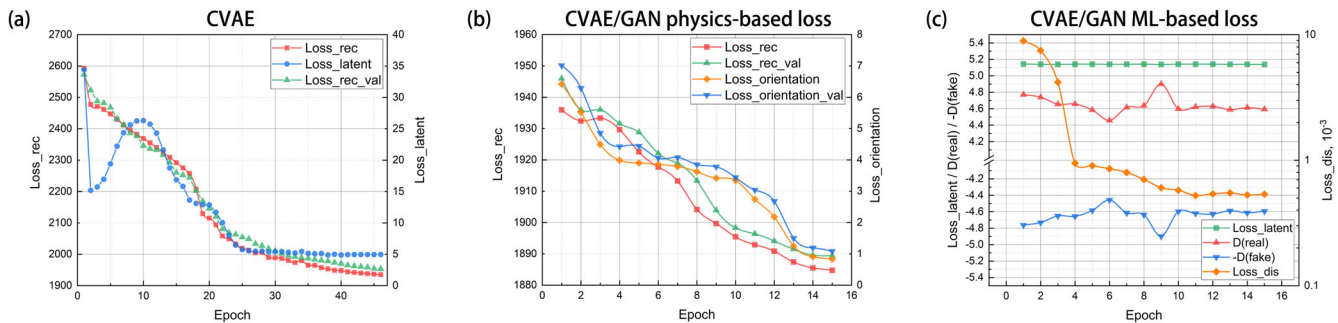


Fig. 2 Training history for different generative models. Training history of: **a** a pure CVAE model, **b** physics-based loss items of a CVAE/GAN model, and **c** ML-based loss items of a CVAE/GAN model ($\text{Loss}_{\text{dis}} = D(\text{real}) - D(\text{fake})$).

the other commonly used mean squared error (MSE), although it shows a better optimization behavior, the loss itself is biased towards 0.5 whenever the ground truth is not binary³⁵. For orientations in the testing data set, the average optimal reconstruction loss for each pattern (60×60 pixels) is 0.503. Currently, for generated patterns, the average reconstruction loss is 0.541.

Since the cross-entropy is not uniform throughout the pixel value range, a more straightforward way is to directly check the pixel value deviation between real and generated patterns. In most studies on generative models, such a pixel-wise comparison is very rare, since location of the features is not a major concern. Figure 3d shows the overall distribution based on the statistics from the whole testing data set, which is a nearly normal fashion; over 80% of the pixels are within 15% from the ground truth.

Finally, the quality of generated patterns is evaluated by the accepted indexing method DI. After applying DI to patterns generated by the trained CVAE model, the disorientation is calculated between the input (i.e., ground truth) orientations and the indexing results of DI. Figure 3e is the original distribution when using a dictionary of $n = 100$ (same density of dictionary for training). The mean and standard deviation are $0.650 \pm 0.257^\circ$. To further get rid of the limitation set by the sampling density of the dictionary, a refinement process is performed based on the BOBYQA (bound optimization by quadratic approximation) optimization algorithm³⁶. The distribution after the refinement is shown in Fig. 3f, with a mean and standard deviation of only $0.142 \pm 0.061^\circ$. Considering the high accuracy and robustness of DI, the extra low average disorientation angle from the orientations input is a powerful endorsement for the pattern quality of the generated patterns.

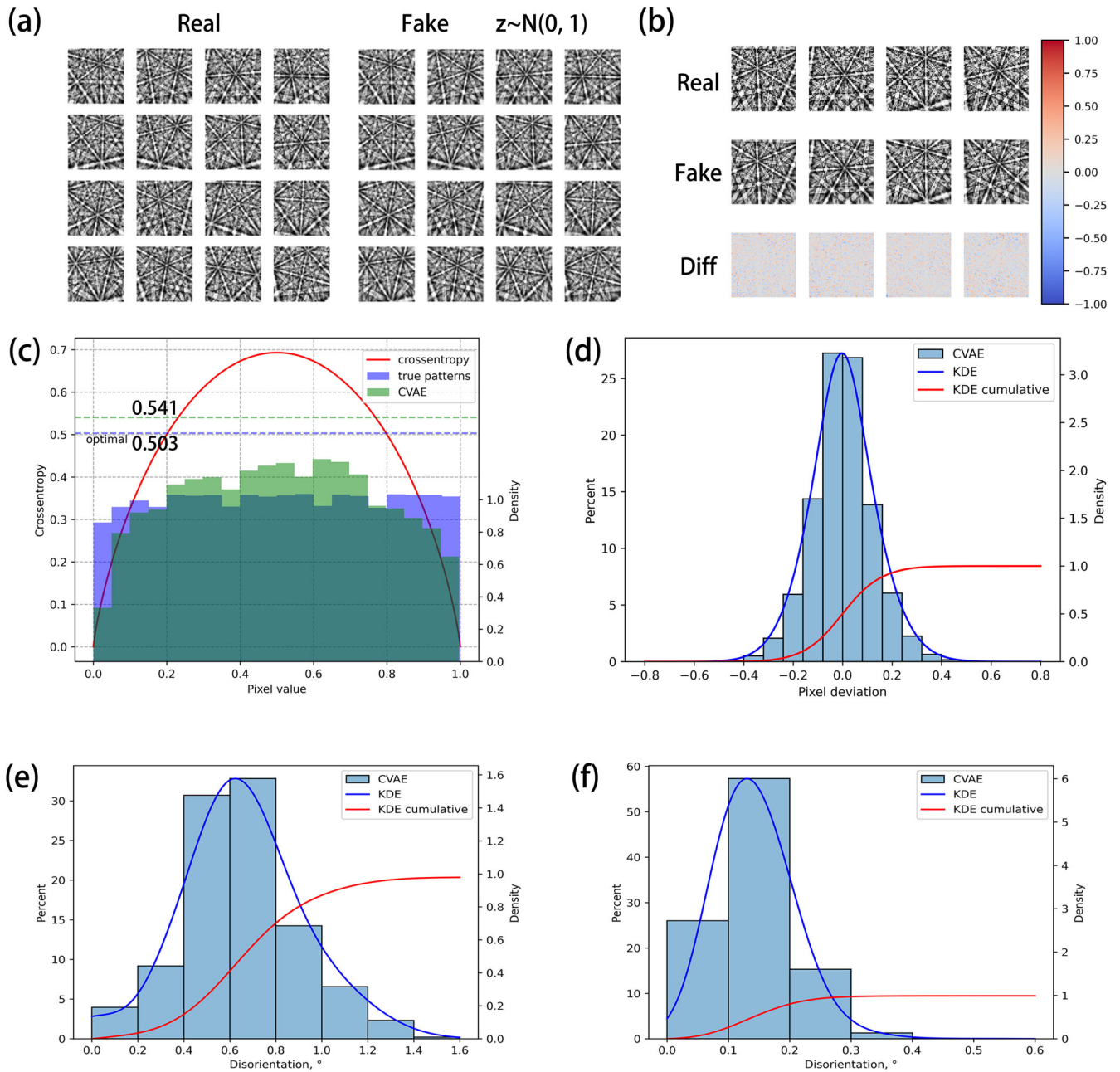


Fig. 3 Quantitative analysis of pattern generated by CVAE model. **a** Generated patterns from a pure CVAE model with adaptive coefficient of KL divergence applied, **b** difference between original and generated patterns, **c** distribution of pixel values in original and generated patterns, **d** distribution of pixel value deviation with kernel density estimation, **e**, **f** distribution of disorientation angle between orientations input and DI results on generated patterns with/without refinement.

Fine tuning through introduction of GAN

To compensate for the problem of blurry edges in patterns generated by the CVAE model, based on the pre-trained discriminator and classifier, the loss items of GAN can be added to the training. Since all components in the model are pre-trained, model collapse is very rare during the fine tuning process. Another important advantage when the loss of GAN is involved is that the orientation, together with other attributes encoded can be directly optimized through items provided by the classifier. When trained on a pure CVAE model, the restoration of EBSD patterns is the priority and there is no item in the loss function directly related to these variables.

From the training history shown in Fig. 2b, c, it can be seen that after the discriminator and classifier are involved, in the fine tuning step both the disorientation loss and reconstruction loss are gradually getting lower, indicating the further improvement in pattern quality; the latent loss remains low at the edge of step function applied as its adaptive coefficient. It is observed that the loss term for discriminator drops quickly in the first several epochs of training and finally stabilizes at a relatively low level, indicating comparable progress of the discriminator and generator during alternating training.

Figure 4a compares patterns generated by the CVAE model and the CVAE/GAN model. As mentioned, before GAN is introduced

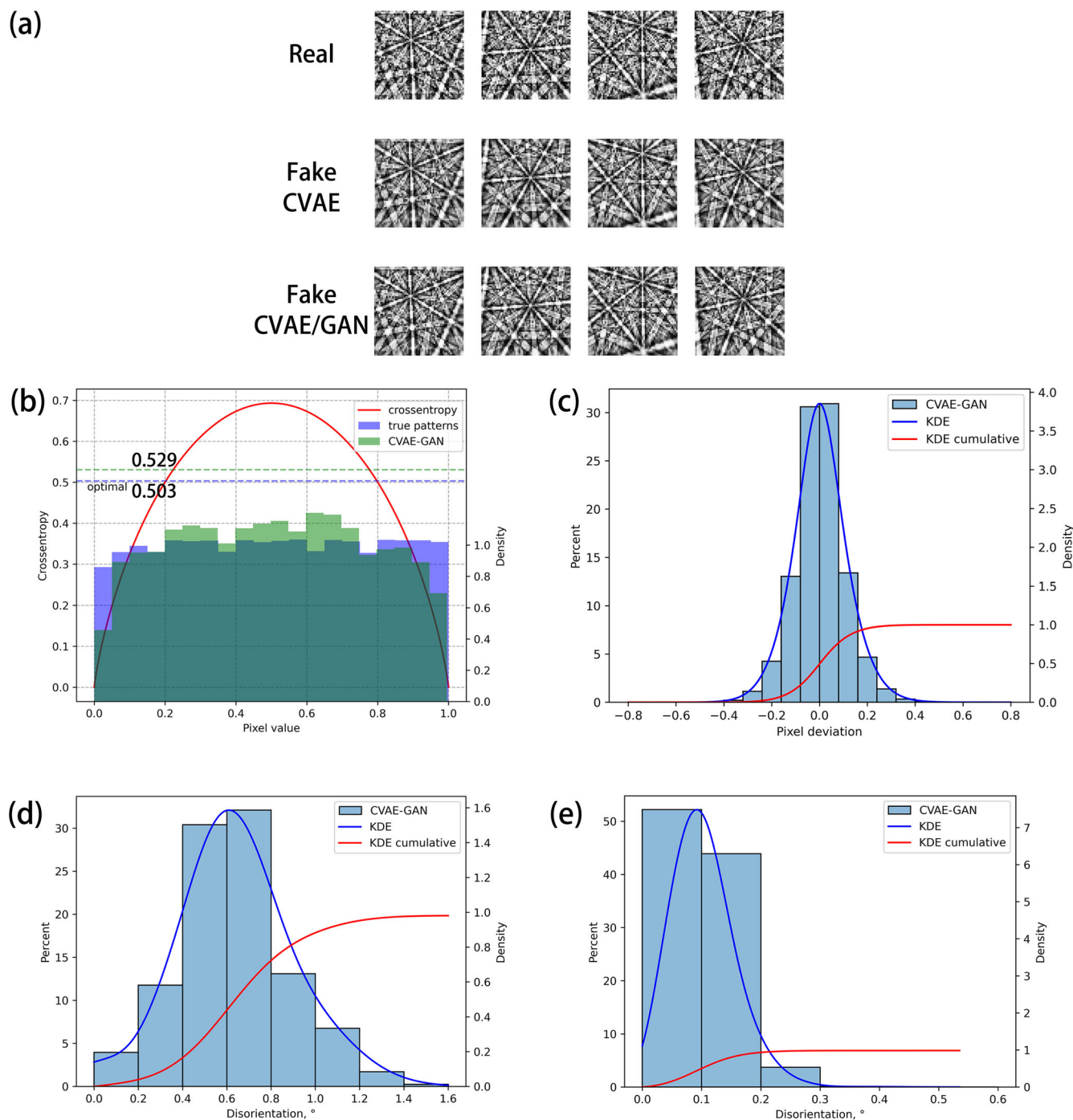


Fig. 4 Quantitative analysis of pattern generated by CVAE/GAN model. **a** Comparison of ground truth, patterns generated by CVAE model, and patterns generated by CVAE/GAN model, **b** distribution of pixel values in original and generated patterns, **c** distribution of pixel value deviation with kernel density estimation, **d**, **e** distribution of disorientation angle between orientations input and DI results on generated patterns with/without refinement.

into the training, the CVAE model can already generate patterns with high quality. Thus the only difference that the naked eye can barely identify from the generated patterns is the mitigation of blurry edges of Kikuchi bands. This is also revealed by the distribution of pixel values (Fig. 4b vs. Fig. 3c), as fewer pixel values are aggregating in the middle of the range. We speculate that this is because the bias towards 0.5 caused by the latent loss is partially balanced by the disorientation loss as well as the discriminator loss.

The improvement brought by GAN is further analyzed with the help of the same quantitative metrics proposed. Figure 4c–e enumerates the distribution from the testing data set and the corresponding statistic is listed in Table 1. Compared with the CVAE model, the distribution of most metrics from the CVAE/GAN model maintains the same trend, but with an obvious progress. Besides the improvement in reconstruction which can be characterized by cross entropy and pixel deviation, the orientation indexed by DI is also closer to the orientation input. Without refinement, two models show almost the same average

disorientation angle, because the resolution of DI is restricted by the sampling density of dictionary, which is the same as the one for training data set. After the indexing result is refined, the difference in accuracy of generating patterns before and after the fine tuning can be observed.

Model performance on multiple manipulated attributes

Having verified the model architecture and training strategy for the orientation-only case, next we include the accelerating voltage in the tensor of manipulated attributes. This increase in dimensionality almost does not change the model size, except for the number of trainable parameters in the last layer of the encoder, the first layer of the decoder/generator, and the last layer of classifier, which directly connect to the attribute tensor. The training history of the model is also very similar to that without the accelerating voltage, as long as each component in the loss function can be balanced to avoid model collapse. The animation in Supplementary Movie 2 indicates a trend similar to that of the model which only takes orientation as the manipulated attribute. Different from Supplementary Movie 1, to better show how the model accommodates to variation introduced by the accelerating voltage, here we focus on EBSD patterns generated with four randomly pick orientations under five different accelerating voltages. Visually, the generated patterns maintain a very high pattern quality. The model is able to isolate the impact of the rising accelerating voltage from that of orientation change, as the location of all Kikuchi bands is maintained, but their widths shrink.

Table 1. The average and standard deviation of quantitative metrics before/after the fine tuning via the introduction of GAN.

Metric	CVAE	CVAE/GAN
Crossentropy	0.541 ± 0.081	0.529 ± 0.065
Absolute pixel deviation	0.092 ± 0.0079	0.077 ± 0.0065
Disorientation angle	0.650° ± 0.257°	0.635° ± 0.253°
Disorientation angle refined	0.142° ± 0.061°	0.102° ± 0.050°

To further quantify the pattern quality in terms of feature location and feature size, we employ once again the DI with refinement algorithm to get the orientation prediction on generated patterns, and then calculate the disorientation angle between it and the corresponding orientation input. Because the accelerating voltage turns into a variable, out of curiosity we also record the result when there is a mismatch between the accelerating voltage set in DI and attribute input; the results are listed in Fig. 5. Each cell is colored from green (low) to red (high) based on its value. It can be seen that disorientation angle is optimal when the indexing accelerating voltage is the same as the one in the attribute input of generative model. Compared with the CVAE/GAN model which only takes orientation as input, the performance here is very close. This demonstrates the great potential and scalability of the model to handle a more complex, multi-dimensional input space. The performance drop caused by the mismatching accelerating voltage for indexing is caused by the change in the width of Kikuchi bands which will lead to the decrease in the dot product value, even if the band center remains the same, and thus it is likely for DI to produce a false prediction.

Finally, to analyze the reaction of the model to varying accelerating voltages, we track the width change of the main Kikuchi bands in both ground truth and generated patterns with a random orientation. It is found that the trend shown in the generated patterns conforms to that in real ones; details are provided in the Supplementary Information.

To sum up, a deep generative model (EBSD-CVAE/GAN) is proposed in this study to realize analytic and parametric EBSD pattern simulation. We demonstrate that with proper training, the model is able to generate EBSD patterns with high fidelity based on the manipulated attributes given. The quality of the generated patterns was evaluated through various means. We also demonstrate that the dimension of the manipulated attributes can be easily extended without much change in model architecture, while maintaining a high pattern quality. As summarized in Table 2, compared with the forward model, which only allows constant value for some parameters at a time, the generative model is more flexible and able to represent a more complete distribution of backscattered electrons. Although extra time is required at the model training stage, it is extremely useful and efficient when multiple parameters are subjected to change in a

Mean disorientation angle after refinement, °

Input av, kV \ Indexing av, kV	10	15	20	25	30
10	0.117	0.163	0.653	0.622	0.813
15	0.155	0.113	0.167	0.156	0.203
20	0.432	0.134	0.111	0.116	0.119
25	0.996	0.242	0.124	0.120	0.122
30	1.266	0.518	0.159	0.153	0.150

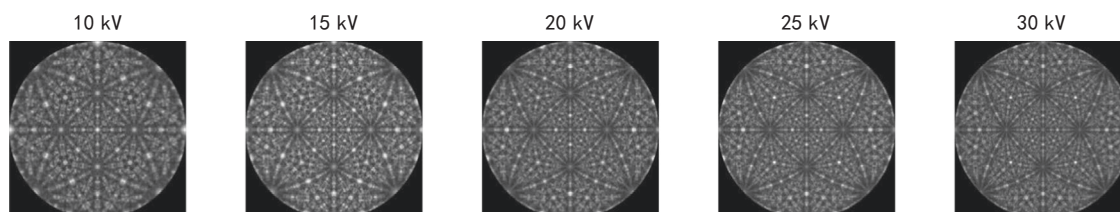
Standard deviation of disorientation angle after refinement

Input av, kV \ Indexing av, kV	10	15	20	25	30
10	0.080	0.097	0.240	0.228	0.258
15	0.093	0.078	0.097	0.094	0.110
20	0.184	0.087	0.080	0.078	0.080
25	0.255	0.125	0.083	0.081	0.081
30	0.278	0.204	0.096	0.093	0.092

Fig. 5 Comparison of input and indexed orientations as a function of accelerating voltage. The refined average and standard deviation of disorientation between orientation input and DI results with different indexing accelerating voltages used.

Table 2. The comparison of forward model and generative model and their usage scenarios.

	Forward model	Generative model
Variables allowed in one model	Orientation and parameters of detector location	All previous, accelerating voltage, etc.
Resources required	Computational	Computational and sufficient patterns for training
Main time consumption	Simulation of “Master” pattern under different configurations	Model training
Accuracy in disorientation (DI with $n = 100$)	$\sim 0.1^{\circ 17}$	0.15°
Usage scenario	Orientation is the only variable; high accuracy required	Multiple variables with large range

**Fig. 6 Master patterns as a function of voltage.** Master patterns of Pt under different accelerating voltages.

single simulation (e.g., simulating EBSD patterns under different accelerating voltages, with a step of 1 kV), or extrapolation from the known variable space is required.

METHODS

Model modules

The design of the encoder is similar to our end-to-end orientation determination model EBSD-CNN¹⁸, consisting of convolutional blocks and fully connected layers. Its main function is to extract features from EBSD patterns for training, and map them into the latent space by outputting parameters of its distribution. The convolution block is composed of depth-wise separable 2D convolution layers with leaky ReLU activation³⁷. A 2D convolution residual block is added to alleviate the vanishing-gradient and overfitting problem in a deep structure by skipping connections.

When the attribute vector is detached from the output of encoder, it is hoped that the rest forms the parameters of an underlying probability distribution. With these parameters of the distribution determined, we can easily sample random latent representations that are ideally separated from the attributes we want to manipulate as part of the input for the decoder/generator. This is also known as the biggest difference between VAE and original autoencoder³⁸. Before being fed into the decoder/generator, the attribute vector is concatenated with the latent representation, forming the composite representation conditioned on the specific attributes. To adjust the number of attributes being manipulated, we only need to change the number of weights in the first layer of the decoder/generator; hence, the model can be flexibly scaled up to control multiple attributes without the need for drastic changes.

The decoder/generator reconstructs EBSD patterns from latent representations with arbitrary attribute vector. Since the representation is conditioned, besides pattern quality, the decoder/generator is also responsible for generating patterns with the correct attributes. In terms of the structure, the biggest difference is that the convolution filters are replaced by deconvolution (transposed convolution) filters in the transposed convolution block. An intuitive description is taking the opposite direction of convolution, which encodes extra information, offers an option to increase the output size, and maintains a connectivity pattern compatible with convolution³⁹.

The discriminator and classifier are inside a single sub-network. After the convolutional blocks for feature extraction from real and generated patterns, they each have fully connected layers to predict the authenticity of patterns and the attributes encoded, respectively. If multiple attributes are manipulated in the latent space, the number of classifiers can be accordingly scaled up, which makes the construction of loss functions for different attributes more convenient. For the detailed configuration of each module used in this study, please refer to “Code availability” section.

Datasets

Initially, only orientation is mutable, thus we generate a equal-volume cubical grid that densely covers the cubic Rodrigues fundamental zone (FZ) through the EMsampleRFZ function in EMsoft⁴⁰. With a sampling density of $n = 100$, in total 333,227 unique orientations are generated with a mean disorientation angle of $\approx 1^{\circ}$. Then, EBSD patterns for all these orientations are simulated via the EMEBSD module¹² to compose the training data set. The datasets for validating and testing are constituted in a similar way, only with different sampling densities so that there is no duplicate in any of them. This avoids potential overfitting and leak of the validating/testing data set during the training phase. In all figures, patterns generated by EMsoft are referred to as “Real,” as opposed to patterns from the generative model with the same orientations and accelerating voltages input marked as “Fake.”

Out of all parameters other than orientation, accelerating voltage is taken to be the extra parameter in this study, because the change in accelerating voltage does not affect the location of the Kikuchi bands, but only their widths. From Fig. 6, it can be seen that with other parameters fixed, a higher accelerating voltage will lead to Kikuchi bands with smaller width. The unchanged center of Kikuchi bands makes it easier to observe whether the model handles this extra parameter well. More importantly, because the changes resulting from the accelerating voltage are relatively small, it is possible to discretize the parameter space, thus avoiding an exponential increase in the size of the training data set. In this study, 5 accelerating voltages are used to cover the range of 10–30 kV, which is used in most EBSD pattern acquisitions.

Loss function

Before decomposing the loss function, we will define the training/testing formulation. Given the pattern \mathbf{x} for training, the encoder G_{enc} will output the parameters $(\boldsymbol{\mu}, \boldsymbol{\sigma})$ of the distribution D in the latent space:

$$(\boldsymbol{\mu}, \boldsymbol{\sigma}) = G_{\text{enc}}(\mathbf{x}) \quad (1)$$

Then, the latent representation is sampled based on the choice of the distribution and its parameters:

$$\mathbf{z} \sim D(\boldsymbol{\mu}, \boldsymbol{\sigma}) \quad (2)$$

Based on the latent representation \mathbf{z} and the attribute vector \mathbf{attr} , the decoder/generator G_{dec} will try to reconstruct the input, i.e., generating “fake” patterns:

$$\hat{\mathbf{x}} = G_{\text{dec}}(\mathbf{z}, \mathbf{attr}) \quad (3)$$

Taking both “real” patterns and “fake” patterns as input, the discriminator G_{dis} and the classifier G_{cls} will predict their authenticity $\hat{\mathbf{y}}$ and manipulated attributes $\hat{\mathbf{attr}}$, respectively:

$$\hat{\mathbf{y}} = G_{\text{dis}}(\mathbf{x}, \hat{\mathbf{x}}) \quad (4)$$

$$\hat{\mathbf{attr}} = G_{\text{cls}}(\mathbf{x}, \hat{\mathbf{x}}) \quad (5)$$

It can be seen that there are multiple inputs and outputs, thus various losses can be configured to guarantee the correct update of weights in each component.

The KL divergence is a type of statistical distance used to measure the difference between two probability distributions⁴¹. A decrease of the KL divergence indicates that the parameterized distribution in the latent space approaches a standard or uniform distribution. Once minimized, the generation of the latent representation can eliminate the real pattern input and the encoder, and only needs to sample from the standard or uniform distribution. Although we define the loss as a KL divergence, when it is intractable, the evidence lower bound (ELBO) is actually optimized during training. For normal distribution, its loss is given by:

$$L_1(\text{Gaussian}) = \log(\boldsymbol{\sigma}^{-1}) + \frac{\boldsymbol{\sigma}^2 + \boldsymbol{\mu}^2}{2} \quad (6)$$

With the latent representation and attribute vector, the decoder/generator should produce realistic images with correct attributes, which are expected to approach the original input of encoder. Thus, the reconstruction learning is set up based on the difference between output of the decoder and input of the encoder:

$$\begin{aligned} L_2 = \ell_{\text{rec}}(\mathbf{x}, \hat{\mathbf{x}}) \\ = \text{Crossentropy}(\mathbf{x}, \hat{\mathbf{x}}) \\ = -\frac{1}{n} \frac{1}{hw} \sum (\mathbf{x} \log \hat{\mathbf{x}} + (1 - \mathbf{x}) \log(1 - \hat{\mathbf{x}})) \end{aligned} \quad (7)$$

where n is the number of images in a training batch and h, w mean height and width of an image respectively. Here the binary cross-entropy is used for better optimization performance. Besides cross-entropy, there are other alternatives, such as mean squared error (MSE). The reconstruction loss helps make the latent representation \mathbf{z} conserve information for the later recovery of the attribute-excluding details.

The attribute classifier constrains the generated patterns to encode the desired attributes. The loss function verifies whether it is able to identify attributes from “real” patterns, as well as the decoder’s capability of generating patterns with the attribute input correctly. Since currently we have orientation and accelerating voltage as variables in the attribute vector, for the former the loss function is the disorientation angle, while for the latter the

loss function is the mean squared error:

$$\text{attr} = (\mathbf{o}, \text{av}) \quad (8)$$

$$L_{3,\mathbf{o}} = \text{Disorientation}(\mathbf{o}, \hat{\mathbf{o}}) \quad (9)$$

$$L_{3,\text{av}} = \text{MSE}(\text{av}, \hat{\text{av}}) \quad (10)$$

The adversarial loss between the generator/decoder and the discriminator is introduced to make the generated patterns visually realistic. Here, we follow the loss function used in WGAN⁴², which characterizes the Wasserstein distance and has its advantages over the Jensen-Shannon divergence (JS divergence) in the original GAN:

$$\begin{aligned} L_{4,\text{dis}} &= -\mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} [D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})} [D(G_{\text{dec}}(\mathbf{z}))] \\ &= -\mathbf{y} + \hat{\mathbf{y}} \end{aligned} \quad (11)$$

$$\begin{aligned} L_{4,\text{dec}} &= -\mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})} [D(G_{\text{dec}}(\mathbf{z}))] \\ &= -\hat{\mathbf{y}} \end{aligned} \quad (12)$$

The first is minimized when training the discriminator, so that it tends to assign a higher probability to the original patterns, while a lower one to generated patterns. The second is minimized when training the decoder/generator, which aims at achieving a higher probability from the discriminator using generated patterns.

To allow for flexibility in the tuning model and to place emphasis on a certain aspect of the generated pattern quality, a coefficient is needed for each loss item mentioned. Thus, the overall loss expression for the encoder and decoder/generator can be written as:

$$L = a_1 L_1 + a_2 L_2 + a_{3,\mathbf{o}} L_{3,\mathbf{o}} + a_{3,\text{av}} L_{3,\text{av}} + a_{4,\text{dec}} L_{4,\text{dec}} \quad (13)$$

As the discriminator and classifier are placed behind the encoder and decoder/generator, the overall loss expression when training them is:

$$L = a_{3,\mathbf{o}} L_{3,\mathbf{o}} + a_{3,\text{av}} L_{3,\text{av}} + a_{4,\text{dis}} L_{4,\text{dis}} \quad (14)$$

Training

The initial training takes the orientation in the form of a quaternion as the only manipulated attribute. To verify the feasibility and determine an appropriate model size, we trained a model with a latent representation size of 500 (i.e., $\mathbf{z} \in \mathbb{R}^{500}$), and set a high ratio for the reconstruction loss, while neglecting the adversarial and attribute losses. This is because mode collapse is common in the training of a GAN when the generator fails to generalize, or the performance gap between the generator and the discriminator keeps expanding, yet sometimes the solution is hard to find⁴³. Thus, we start from a pure CVAE to validate the encoder-decoder architecture in this study. To balance the reconstruction loss and KL divergence, we applied an adaptive coefficient in the final loss function (detail discussed in the Supplementary Information).

Both discriminator and classifier are essentially end-to-end discriminative models, thus they are pre-trained using the same recipe as the one for EBSD-CNN, our orientation determination model¹⁸. After CVAE part and discriminator/classifier are trained separately, they are integrated and trained in parallel following a specific frequency as a CGAN model. Weight clipping with constant minimum/maximum is used to enforce a Lipschitz constraint and maintain a stable training of the model. It is expected that through adversarial training both parts can gain further progress at a similar pace.

All hyperparameters used in different sections throughout the training are provided in the Supplementary Information.

DATA AVAILABILITY

The data that support the findings of this study for model training, and testing can be generated via the open source package EMsoft following the instructions in "Datasets" under the "Methods" section. It is also available on request from the corresponding author M.D.G.

CODE AVAILABILITY

The source code to train the generative model EBSD-CVAE/GAN is provided open source under non-restrictive licenses. The source code, together with the documentation, can be obtained from the GitHub repository: https://github.com/Darkhunter9/EBSD_CVAE_GAN_Public.

Received: 25 May 2022; Accepted: 23 September 2023;

Published online: 23 October 2023

REFERENCES

- Nishikawa, S. & Kikuchi, S. Diffraction of cathode rays by calcite. *Nature* **122**, 726 (1928).
- Schwarzer, R. A., Field, D. P., Adams, B. L., Kumar, M. & Schwartz, A. J. *Present State of Electron Backscatter Diffraction and Prospective Developments* 1–20 (Springer US, 2009).
- Wright, S. I. & Adams, B. L. Automatic analysis of electron backscatter diffraction patterns. *Metall. Trans. A* **23**, 759–767 (1992).
- Lassen, N. K., Jensen, D. J. & Conradsen, K. Image processing procedures for analysis of electron back scattering patterns. *Scanning Microsc.* **6**, 115–121 (1992).
- Hough, P. V. Method and means for recognizing complex patterns. US patent 3,069,654 (1962).
- Calcagnotto, M., Ponge, D., Demir, E. & Raabe, D. Orientation gradients and geometrically necessary dislocations in ultrafine grained dual-phase steels studied by 2D and 3D EBSD. *Mater. Sci. Eng. A* **527**, 2738–2746 (2010).
- Kamaya, M. Assessment of local deformation using EBSD: Quantification of local damage at grain boundaries. *Mater. Charact.* **66**, 56–67 (2012).
- Britton, T. B. & Hickey, J. L. Understanding deformation with high angular resolution electron backscatter diffraction (HR-EBSD). *IOP Conf. Ser. Mater. Sci. Eng.* **304**, 012003 (2018).
- Nowak, W. J. The use of ion milling for surface preparation for EBSD analysis. *Materials* **14**, 3970 (2021).
- Winkelmann, A., Trager-Cowan, C., Sweeney, F., Day, A. & Parbrook, P. Many-beam dynamical simulation of electron backscatter diffraction patterns. *Ultramicroscopy* **107**, 414–421 (2007).
- Callahan, P. G. & De Graef, M. Dynamical electron backscatter diffraction patterns. Part I: Pattern simulations. *Microsc. Microanal.* **19**, 1255–1265 (2013).
- Singh, S., Ram, F. & De Graef, M. EMsoft: open source software for electron diffraction/image simulations. *Microsc. Microanal.* **23**, 212–213 (2017).
- Chen, Y. H. et al. A dictionary approach to electron backscatter diffraction indexing. *Microsc. Microanal.* **21**, 739–752 (2015).
- De Graef, M. A dictionary indexing approach for EBSD. *IOP Conf. Ser. Mater. Sci. Eng.* **891**, 012009 (2020).
- Lenthe, W. C., Singh, S. & Graef, M. D. A spherical harmonic transform approach to the indexing of electron back-scattered diffraction patterns. *Ultramicroscopy* **207**, 112841 (2019).
- Hielscher, R., Bartel, F. & Britton, T. B. Gazing at crystal balls - electron backscatter diffraction indexing and cross correlation on a sphere. *Microsc. Microanal.* **25**, 1954–1955 (2019).
- Ram, F., Wright, S., Singh, S. & De Graef, M. Error analysis of the crystal orientations obtained by the dictionary approach to EBSD indexing. *Ultramicroscopy* **181**, 17–26 (2017).
- Ding, Z., Pascal, E. & De Graef, M. Indexing of electron back-scatter diffraction patterns using a convolutional neural network. *Acta Mater.* **199**, 370–382 (2020).
- Ding, Z., Zhu, C. & De Graef, M. Determining crystallographic orientation via hybrid convolutional neural network. *Mater. Charact.* **178**, 111213 (2021).
- Kaufmann, K. et al. Crystal symmetry determination in electron diffraction using machine learning. *Science* **367**, 564–568 (2020).
- Kaufmann, K. et al. Phase mapping in EBSD using convolutional neural networks. *Microsc. Microanal.* **26**, 458–468 (2020).
- Heigold, G., Ney, H., Lehnen, P., Gass, T. & Schluter, R. Equivalence of generative and log-linear models. *IEEE Trans. Audio Speech Lang. Process.* **19**, 1138–1148 (2010).
- Nguyen, A., Clune, J., Bengio, Y., Dosovitskiy, A. & Yosinski, J. Plug & play generative networks: conditional iterative generation of images in latent space. In

- Proc. 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017* 3510–3520 (IEEE, 2016).
- Hong, S., Yang, D., Choi, J. & Lee, H. Inferring semantic layout for hierarchical text-to-image synthesis. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition 7986–7994* (IEEE, 2018).
- He, Z., Zuo, W., Kan, M., Shan, S. & Chen, X. AttGAN: facial attribute editing by only changing what you want. *IEEE Trans. Image Process.* **28**, 5464–5478 (2019).
- Karacan, L., Akata, Z., Erdem, A. & Erdem, E. Learning to generate images of outdoor scenes from attributes and semantic layouts. Preprint at <https://arxiv.org/abs/1612.00215> (2016).
- Yao, S. et al. 3D-aware scene manipulation via inverse graphics. In *Advances in Neural Information Processing Systems* 1887–1898 (NIPS, 2018).
- Kalinin, S. V., Dyck, O., Jesse, S. & Ziatdinov, M. Exploring order parameters and dynamic processes in disordered systems via variational autoencoders. *Sci. Adv.* **7**, eabd5084 (2021).
- Ziatdinov, M. & Kalinin, S. Robust feature disentanglement in imaging data via joint invariant variational autoencoders: from cards to atoms. Preprint at <https://arxiv.org/abs/2104.10180> (2021).
- Kingma, D. P. & Welling, M. Auto-encoding variational Bayes. Preprint at <https://arxiv.org/abs/1312.6114v10> (2013).
- Goodfellow, I. et al. Generative adversarial networks. *Commun. ACM* **63**, 139–144 (2014).
- Sohn, K., Lee, H. & Yan, X. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems* 28 (NIPS, 2015).
- Mirza, M. & Osindero, S. Conditional generative adversarial nets. Preprint at <https://arxiv.org/abs/1411.1784> (2014).
- Pizer, S. M. et al. Adaptive histogram equalization and its variations. *Comput. Vis. Graph. Image Process.* **39**, 355–368 (1987).
- Creswell, A., Arulkumar, K. & Bharath, A. A. On denoising autoencoders trained to minimise binary cross-entropy. Preprint at <https://arxiv.org/abs/1708.08487> (2017).
- Powell, M. *The Broyden Algorithm for Bound Constrained Optimization Without Derivatives*. Technical Report, Department of Applied Mathematics and Theoretical Physics (2009).
- Xu, B., Wang, N., Chen, T. & Li, M. Empirical evaluation of rectified activations in convolutional network. Preprint at <https://arxiv.org/abs/1505.00853> (2015).
- Doersch, C. Tutorial on variational autoencoders. Preprint at <https://arxiv.org/abs/1606.05908> (2021).
- Dumoulin, V. & Visin, F. A guide to convolution arithmetic for deep learning. Preprint at <https://arxiv.org/abs/1603.07285?context=cs> (2016).
- Roşca, D., Morawiec, A. & De Graef, M. A new method of constructing a grid in the space of 3D rotations and its applications to texture analysis. *Model. Simul. Mat. Sci. Eng.* **22**, 075013 (2014).
- Kullback, S. & Leibler, R. A. On information and sufficiency. *Ann. Math. Stat.* **22**, 79–86 (1951).
- Arjovsky, M., Chintala, S. & Bottou, L. Wasserstein GAN. Preprint at <https://arxiv.org/abs/1701.07875> (2017).
- Arjovsky, M. & Bottou, L. Towards principled methods for training generative adversarial networks. Preprint at <https://arxiv.org/abs/1701.04862> (2017).

ACKNOWLEDGEMENTS

The authors gratefully acknowledge funding from a DoD Vannevar-Bush Faculty Fellowship (N00014-16-1-2821), as well as the computational facilities of the Materials Characterization Facility at CMU under grant # MCF-677785. Use was made of computational facilities purchased with funds from the National Science Foundation (grant # 1925717: CC* Compute: A high-performance GPU cluster for accelerated research) and administered by the Center for Scientific Computing (CSC). The CSC is supported by the California NanoSystems Institute and the Materials Research Science and Engineering Center (MRSEC; NSF DMR 1720256) at UC Santa Barbara.

AUTHOR CONTRIBUTIONS

Z.D. generated the CVAE/GAN model, performed the training and the subsequent data analysis; Z.D. wrote and M.D.G. reviewed/edited the manuscript.

COMPETING INTERESTS

The authors declare no competing interests.

ADDITIONAL INFORMATION

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s41524-023-01143-z>.

Correspondence and requests for materials should be addressed to Marc De Graef.

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

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



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

© The Author(s) 2023