Towards automatic feature extraction and sample generation of grain structure by variational autoencoder

Ye Ji [a,b,c,d,1], Arnd Koeppe [d,e,1,2,*], Patrick Altschuh [d,f], Deepalaxmi Rajagopal [d], Yinghan Zhao [d], Weijin Chen [a,b,c], Yi Zhang [a,b,c], Yue Zheng [a,b,c], Britta Nestler [d,e,f]

[a] Centre for Physical mechanics and Biophysics, School of Physics, Sun Yat-sen University, Xin'gang West Road 135, Guangzhou, 510275, Guangdong, China
[b] Guangdong Provincial Key Laboratory of Magnetoelectric Physics and Devices, School of Physics, Xin'gang West Road 135, Guangzhou, 510275, Guangdong, China
[c] State Key Laboratory of Optoelectronic Materials and Technologies, School of Physics, Xin'gang West Road 135, Guangzhou, 510275, Guangdong, China
[d] Institute of Nanotechnology (INT-MSS), Karlsruhe Institute of Technology, Kaiserstraße 12, Karlsruhe, 76131, Baden-Württemberg, Germany
[e] Institute for Applied Materials (IAM-MMS), Karlsruhe Institute of Technology, Kaiserstraße 12, Karlsruhe, 76131, Baden-Württemberg, Germany
[f] Institute for Digital Materials Science, Karlsruhe University of Applied Sciences, Moltkestr. 30, Karlsruhe, 76133, Baden-Württemberg, Germany

ARTICLE INFO

Keywords:
Grain structure
Machine learning
Variational autoencoder

ABSTRACT

Granularity is one of the most predominant structures for polycrystalline materials, from geological compounds to technical high-performance alloys. In particular, grain structures in metal alloys enable custom-tailored material properties if their formation can be modeled and simulated accurately to analyze, characterize, and generate granular materials. Despite significant advances in the field, established physics-based grain-growth models still need to improve to match real-life experimental data accurately. Instead of relying on physical knowledge, data-driven models provide a complementary way to study grain structures using only the information in the data. However, many machine learning methods require supervised learning on information that is challenging to measure or numerically expensive to compute. In this work, we propose a new strategy that uses unsupervised deep learning on synthetic physics-based data to generate realistic granular structures with customizable features and properties (grain size/number). A variational autoencoder learns to compress grain structures into a low-dimensional latent space, derive data-driven features that characterize grain structures, and use these features to generate new structures with representative grain size distributions in a computationally efficient way. These data-driven features can complement the physics-based features derived from classical materials research and help identify new characterizing properties. Thus, this work represents a prototype application of bridging machine learning methods to material characterization and generation.

1. Introduction

Granular structures, a compact packing configuration of multiple cells, can be observed in many natural systems, such as dried basins of lakes, metal alloys, foams, geological rocks or wings of insects. These structures continuously inspire humans to create stunning architectures and artworks, like Water Cube and Airspace Tokyo. The grain structures in the metal alloy are granular systems of significant importance in material sciences, where the grains represent small single crystalline regions, and the borders separating them are called grain boundaries. The spatial structure of the grains strongly affects material properties and behavior. Particularly, the knowledge of correlations between grain structure characteristics and material properties allows for the development of new materials and processing routes based on advanced data analysis techniques in an acceleration mode. For example, the Hall–Petch relation [1] describes that yield strength is inversely dependent on grain size. Furthermore, thermal conductance [2] and electrical conductivity [3] are known to be regulated by grain structure as well. So far, grain structure has been a fascinating area to explore the structure–property linkage in materials science.

Intuitively, the grains interact, which means the contacting interfaces and angles between grains are under certain constraints. The

* Corresponding author at: Centre for Physical mechanics and Biophysics, School of Physics, Sun Yat-sen University, Xin'gang West Road 135, Guangzhou, 510275, Guangdong, China.
E-mail addresses: jiye5@mail2.sysu.edu.cn (Y. Ji), arnd.koeppe@kit.edu (A. Koeppe).
[1] These authors contribute equally to this work.
[2] Permanent address: Institute for Applied Materials (IAM-MMS), Karlsruhe Institute of Technology, Kaiserstraße 12, Karlsruhe, 76131, Baden-Württemberg, Germany.

constraints could be classified into two aspects. On the one hand, since the total spatial volume remains constant, a particular grain will grow only if the others are shrinking; this effect is geometrical. On the other hand, since a particular grain is surrounded by its neighbors, the faces and vertices of adjacent grains must be paired and tangential; this effect is topological. Reasonable grain structures must fulfill these two criteria. Following this, various analytical theories are built. Von Neumann first proposed the N-6 rule [4] of grain structure based on pure topological considerations. Mullins proposed an idealized model where the moving speed of the grain boundary is proportional to its curvature [5]. Later, Hillert developed a theory [6] for predicting grain size distribution in equilibrium. Subsequent studies extensively explored the possibility of using different state variables (e.g. volume [7], topological class [8,9], topological events [10], etc.) to mathematically describe grain growth behavior.

Besides the analytic theories, several computational methods have been developed to mimic grain growth in last decades, including cellular automata [11], Monte Carlo Potts model [12], and phase-field modeling [13] and multi-phase field modeling [14]. Within these methods, all the discrete elements in the computational domain evolve following some rules, and jointly form a realistic grain configuration after a period of time. The rules used in the model can incorporate physical interactions and therefore can be more realistic. The computational methods depend on classical theories to check their validity. In practice, the Hillert theory is often used to check computer simulation results [13,15,16]. Usually, satisfying matches can be observed between simulation and theory, so the above simulation methods become the gold standard for obtaining large-scale grain structures. Moreover, these simulation methods are further used to validate newly proposed kinetic laws of grain growth [8,17–19] and extended to more complicated systems [20].

Remarkable successes are made so one can generate grain structures without much coding effort and evaluate the analytic laws. Nevertheless, this generation-evaluation workflow confronts ever-increasing challenges. For the simulation side, almost all algorithms rely on the curvature-driven rule, whereas experimental evidence does not support this rule [21,22]. For the evaluation side, whether the Hilbert self-similar regime exists has been questioned [23–25] for a long time. From a physical point of view, the formation of grain structure results from the competitive growth of multiple single crystalline regions. Considering that a physical system favors a low-energy state, in this scenario, the interplay of surface energy and bulk energy plays an important role. Therefore, it is theoretically feasible to obtain a realistic grain structure by incorporating all possible state variables into the system's free energy. Yet, the formulation of energy functional is not trivial, often requiring sophisticated knowledge of the underlying physical phenomena. Specifically, grain growth not only undergoes short-range interactions, such as strain induced by surface stress or latent heat but also possibly possesses long-range interactions, such as magnetic or electrostatic interactions. Therefore, alternative methods to characterize and generate grain structures, tackling complexity with incomplete information, are urgently needed.

With the rapid development of data science, huge progress has been achieved in applying machine learning techniques in materials science [26–29]. These studies cover a wide range of topics, from structure characterization and reconstruction to structure–property linkage. Neural network based models are reported to segment rock images [30] and to reconstruct oolitic limestone [31]. An open-source workflow is built for the comprehensive characterization of porous materials [32]. The optimization of mechanical properties is reported via high-throughput phase-field simulation combined with an artificial neural network [33]. The effective permeability of membranes is found to rely on the pore-solid interface [34]. A regression model is formulated to predict the sluggish effect in duplex and triplex systems [35]. The long short-term memory network is useful for accelerating simulations [36]. We can find that convolutional neural networks are fundamental building blocks in most spatial-data-oriented studies. It is not surprising because CNN can learn to detect the salient features of spatial input data hierarchically without requiring any prior knowledge about physical behavior [37]. When the input data is projected into a low-dimensional space, analyzing the structure–property linkage yields many benefits. Meanwhile, the compressed information is a good starting point to generate new samples.

The analysis above suggests that the Variational AutoEncoder (VAE) [38] is a promising architecture to characterize and generate grain samples. VAE is one of the powerful unsupervised machine learning algorithms derived from the family of variational Bayesian methods, which can be implemented as a neural network built on top of many convolution blocks. The model maps the input data through a reduced-order bottleneck back onto itself, thus maximizing the information compressed into the latent variables. Once the training converges, the encoder learns to characterize a structure, and the decoder learns to generate a structure from a characterization. VAE was proven to be useful for the generation of stochastic porous structures [39]. Yet, a discussion is still absent for the more regular-shaped grain structure. A closely related work is from Yan et al. [40], where they successfully predicted grain growth after learning the local grain boundary behaviors. In comparison, we focus on the feasibility of a purely data-driven workflow. Note that generative adversarial networks can also generate realistic structure [41], but it lacks the ability to characterize structures.

In this paper, we thus select the VAE as our machine learning model, showing that VAE can extract the main feature of the grain structure and generate new grain samples from customized vectors. The conceptual design of VAE is illustrated in Fig. 1. From the left to the right, it is usually called the encoder, the latent space, and the decoder, respectively. First of all, to generate sufficient amounts of data, we simulate multi-phase field models, which have been widely used in recent studies [16,20,25,42]. Built on appropriate data preparation like thresholding and denoising and a programming interface for converting data, the model should also works on experimental data. Then, we build a workflow to convert and preprocess the simulation data into samples of numerical tensors for machine learning. Next, a VAE is trained until it can map these numerical tensors back onto themselves with only a minor reconstruction error. After that, a watershed segmentation yields the discrete grain structure from the reconstructed data. We found that the compressed data in the low-dimensional latent space forms a regular manifold, which can be efficiently explored with a principal component analysis, which uncovers a strong influence of grain-structure properties on the first principal component. In addition, we reveal that the output of the decoder is the weighted sum of a series of eigenmodes. Finally, we generate new grain samples based on the statistical feature of the training dataset, whose statistical feature does not exist in the original training dataset.

2. Methods

2.1. Multi-phase field simulation

In a poly-crystalline material system with N grains, an α grain is treated by an order parameter, so called phase field variable $\phi_\alpha(\mathbf{x}, t)$, occupying spatial–temporal coordinates $\mathbf{x}$ and t, so that the whole system could be represented as a vector-valued order parameter $\phi(\mathbf{x}, t) = (\phi_1, \phi_2, \phi_3, ..., \phi_N)$. Tracking the evolution of this vector valued field variable gives all the information on grain growth. For example, the volume of a particular grain α is obtained by integrating the selected phase field variable $\phi_\alpha(\mathbf{x}, t)$ over all spatial coordinates. The thermodynamic theory states that the system evolves towards the minimum free energy. Following Nestler et al. [14,42], the free energy could be written as

$$\mathcal{F}(\phi) = \int_\Omega \epsilon a(\phi, \nabla\phi) + \frac{1}{\epsilon} w(\phi) dx \tag{1}$$

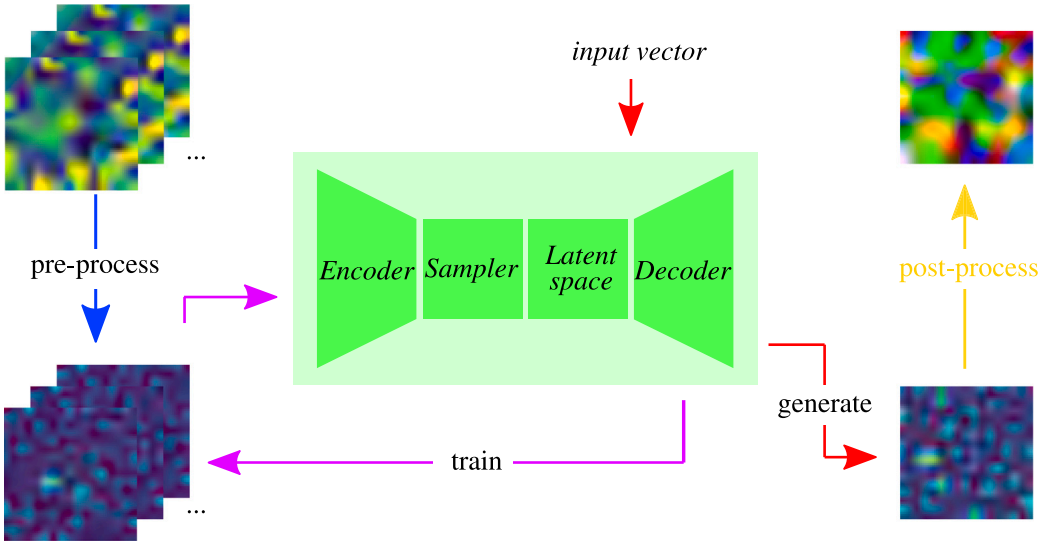

Fig. 1. Overview of the workflow. A batch of index data (upper-left panel) is converted into distance data (lower-left panel), which are then loaded into the variational autoencoder for model training. After convergence is achieved, a customized vector can be used to generate new distance data and corresponding index data.

where $a(\phi, \nabla\phi)$ stands for the surface energy due to the existence of phase boundaries, $w(\phi)$ is the multi-obstacle potential, and ϵ is a factor determining the thickness of diffuse boundary layers. The free energy integrates over the whole volume Ω.

The surface energy depends solely on the phases that form the surfaces. Without loss of generality, it takes the following form of summation of all possible two-phase pairs.

$$a(\phi, \nabla\phi) = \sum_{\alpha<\beta} \gamma_{\alpha\beta}[a_{\alpha\beta}(q_{\alpha\beta})]^2 |q_{\alpha\beta}|^2 \tag{2}$$

where $q_{\alpha\beta}$ is the generalized gradient of phase α and phase β, i.e., $q_{\alpha\beta} = \phi_\alpha \nabla\phi_\beta - \phi_\beta \nabla\phi_\alpha$, and the quadratic form of $q_{\alpha\beta}$ multiplying $\gamma_{\alpha\beta}$ gives a measure of energy. Additionally, $a_{\alpha\beta}$ is an anisotropic factor. It regulates the energy, which depends on $q_{\alpha\beta}$ as well.

The multi-obstacle function is given by the summation of polynomials of order parameters

$$w(\phi) = \frac{16}{\pi^2} \sum_{\alpha<\beta} \gamma_{\alpha\beta}\phi_\alpha\phi_\beta + \sum_{\alpha<\beta} \gamma_{\alpha\beta\delta}\phi_\alpha\phi_\beta\phi_\delta \tag{3}$$

where the quadratic term forces the order parameter to be 1 within the grain and 0 outside the grain, and the cubic term is introduced to suppress the occurrence of extraneous contributions of phase field variables.

Following Landau's formulation, a variational differentiation of the free energy functional with respect to the order parameters ϕ_α yields a set of coupled nonlinear partial differential equations describing the evolution of the grain structure in space and time.

$$\tau\epsilon \frac{\partial\phi_\alpha}{\partial t} = \epsilon\left[\nabla \cdot a_{,\nabla\phi_\alpha}(\phi, \nabla\phi) - a_{,\phi_\alpha}(\phi, \nabla\phi)\right] - \frac{1}{\epsilon} w_{,\phi_\alpha}(\phi) - \lambda \tag{4}$$

where the $(\cdot_{,\phi_\alpha})$, $(\cdot_{,\nabla\phi_\alpha})$ denote partial derivatives. τ is a kinetic factor affecting the velocity of evolution,

$$\tau = \frac{\sum \phi_\alpha\phi_\beta\tau_{\alpha\beta}}{\sum \phi_\alpha\phi_\beta}. \tag{5}$$

The parameter λ computes the impact of all grains. λ ensures that the constraint $\sum_{\alpha=1}^{N} \phi_\alpha(\mathbf{x}, t) = 1$ and takes the following expression,

$$\lambda = \frac{1}{N} \sum_{\alpha=1}^{N} \left[\epsilon\left[\nabla \cdot a_{,\nabla\phi_\alpha}(\phi, \nabla\phi) - a_{,\phi_\alpha}(\phi, \nabla\phi)\right] - \frac{1}{\epsilon} w_{,\phi_\alpha}(\phi)\right]. \tag{6}$$

In the phase field simulation of grain boundary evolution, material parameters from nickel alloys are chosen as reported in [43]. For simplicity, anisotropy is neglected, and the kinetic factor is set to 1. Periodic domain boundary conditions are assumed along the x and y axis.

2.2. Principal component analysis

Principal Component Analysis (PCA) stands as a potent mathematical technique essential for dimensionality reduction and data compression while preserving the intrinsic patterns inherent within the data. PCA's core principle centers on the transformation of the original data into a novel coordinate system where data variability is maximally captured.

The initial step in PCA involves a dataset represented as a matrix, denoted as $X_{n,p}$, where n signifies the number of observations and p represents the number of features. This process commences with data centering, whereby the mean of each feature is subtracted from its respective column, resulting in a demeaned matrix that removes any bias in the data. Subsequently, PCA delves into the identification of orthogonal eigenvectors that encapsulate the maximum variance within the demeaned data. These eigenvectors, when sorted by eigenvalues representing their variance contribution, become the axes of a fresh coordinate system. The leading eigenvector corresponds to the direction with the highest variance, followed by successive eigenvectors capturing progressively less variance. Mathematically, the original dataset $X_{n,p}$ can be expressed as a linear combination of these eigenvectors $V_{p,k}$ and their corresponding scores $Y_{n,k}$:

$$X_{n,p} = Y_{n,k} \cdot V_{p,k}^T, \tag{7}$$

where the variable k signifies the number of considered principal components. The selection of primary components based on their scores allows for effective data dimensionality reduction while preserving the most salient patterns. Additionally, PCA facilitates data reconstruction from the reduced PCA space. This process entails multiplying PCA coordinates $Y'_{n,k}$ by their corresponding eigenvectors $V_{p,k}$ and subsequently adding column-wise mean values to obtain high-dimensional counterparts. The PCA implementation used in this work is based on the scikit-learn package [44].

2.3. Watershed segmentation

The watershed algorithm treats 2D data as a geographical land scope. Firstly, one has to mark the peaks where the local maximum

resides. The calculation of the gradient vectors of the distance data provides this information. Then these peaks are trimmed to remove saddle points [45]. The remaining peaks are the starting points to form islands by gradually lowering the water level until adjacent islands contact each other. The numbers of the islands are given by the peaks, and the neighborhood relationships of the islands are recorded during their formation.

3. Results

3.1. Preparation of training dataset

Since the model should represent a single compound, we simulate this compound as the 2D grain growth process with the multi-phase field method, performed with the simulation framework PACE3D [46], developed at the IDM at the Karlsruhe University of Applied Sciences. The theory of multi-phase field simulation is briefly described in the following. The simulation domain size is 4000 × 4000 numerical cells to finely resolve a 1 μm × 1 μm area, which is filled with 60 000 randomly distributed grains by Voronoi tessellation. Such settings enable getting as many representative configurations as possible. The simulation results are recorded at specific time steps that characterize the physical grain growth process at early (1st), intermediate (2nd), and late (3rd) stages. The initial and final few time steps are not recorded. The initial step refers to a Voronoi partition, which usually deviates a lot from the physical state of grain boundaries as grain boundary energies and curvature minimization are not included in the generation algorithm [47]. Furthermore, the final steps lose statistical significance as the structure may contain only a few residual grains after coarsening.

Using equilibrium or quasi-equilibrium states offers several advantages. First and foremost, these states are common occurrences in the process of grain growth. Moreover, they maintain the homogeneity of the material, making the behavior more predictable in daily use. Secondly, in the context of equilibrium or quasi-equilibrium states, the average size or the number of grains as a scalar value serves as a well-defined feature. This feature is favorable during the evaluation of the generated samples. While the present study primarily focuses on equilibrium or quasi-equilibrium states, it is important to note that non-equilibrium states can also be generated using our multiphysics framework. This suggests that the workflow has the potential to be extended to more intricate scenarios in future endeavors.

A typical computationally generated grain structure is illustrated in Fig. 2a, where each grain is indexed by an integer and represented by a colored area. We note that this index data is unsuitable for model training because the grain index is interchangeable without changing any physical state. For example, after the disappearance of the grain during evolution, the corresponding grain index is missing. Then, the whole data range of indices is neither continuous nor constant. This is detrimental for a neural network to learn the characteristics. Therefore, we convert the index data into binary data, as shown in Fig. 2b, where only the grain boundaries are kept. The binary data preserves all information of grain states, and all pixel values are reduced to zeros or ones, thus overcoming the drawback of index data. However, the sparsity of the boundaries between the grains creates an imbalance in the dataset between the two classes in the binary dataset. Furthermore, within a grain, all values are identical, which means the distance information of a point from the grain boundary is not explicitly carried. Although the distance information is deterministic when the whole structure is given, it is not the same situation where convolutional operations are performed with small filter sizes in a neural network. Therefore, we calculate the Euclidean distance transformation of the binary data and obtain another form of data, hereinafter referred to as distance data (Fig. 2c). This distance data inherently encodes information regarding the shape of the entire grain locally to each material point. Through the distance transformation, the resulting data are less sparse than the binary data, which mitigates the binary class imbalance by reformulating the classification to a regression task, and augments the information contained at every spatial position.

To make our model computationally affordable, we split the distance data into many small patches, where each patch is called a sample. Each sample size is 256 × 256, which is a common dimensionality for computer vision tasks. We can use a single simulation best by sampling the distance data repeatedly rather than simulating numerous small-scale grain growth processes. The sampling details are given in Fig. 2d, where highlighted squares are data to be taken, and red rectangles are the bounding box defined on a global coordinate. The position of each square is randomly chosen, specified by a coordinate (x_i, y_i) located at its upper-left corner. The orientation of each square is allowed to change to increase the variance of data. The size of the bounding box L_b can be calculated analytically based on edge length L_d and orientation angle α_i. Overlapping of patches is allowed. Once a bounding box is sliced from the whole distance data, rotation and cropping are applied to get a sample with the desired size. The relative geometrical relationship is depicted in Fig. 2e. For each simulation step, we pick up 20 000 samples. Three simulation steps are used in the following study. We further use the tfrecords [48] format to store the samples, accelerating the loading speed in model training. As shown in Fig. 2f, the corresponding simulation step for each sample, as descriptors, is also added to the tfrecords.

3.2. Training the variational autoencoder

Variational autoencoders are powerful generative models based on Bayesian interference. Let us define a sample $\mathbf{x}$, The evidence probability of data $p(\mathbf{x})$ is proven to have a lower bound, which means maximization of the evidence lower bound (ELBO) gives an estimation of $p(\mathbf{x})$. In practice, the posterior probability $p(\mathbf{z}|\mathbf{x})$ and prior probability $p(\mathbf{z})$ are taken as Gaussian distribution to make it mathematically simple, where z is the variable and p(z) denotes the probability density of z. In Fig. 3a, the architecture of a VAE is illustrated. The left part with warm colors is the encoder, functioning as the posterior probability $p(\mathbf{z}|\mathbf{x})$. At the center, random sampling reparameterizes the encoder output and applies noise so that the right part, i.e., the decoder, approximates the likelihood probability $p(\mathbf{x}|\mathbf{z})$. The VAE model is implemented with the CIDS framework, an open-source toolbox that automatizes hyperparameter search, model training, and inference aimed at data for the physical and materials sciences [49]. The encoder consists of three layers; each consists of two convolution layers with rectified linear activation followed by a pooling layer. The filter is 2 dimensional with its size being 3 × 3 and the number of filters is 32, 64, and 128, respectively. The stride is chosen to be 1. The pooling size is set to be 4 for anti-aliasing. An additional batch normalization layer is appended after each convolution. The decoder mirrors the architecture of the encoder. The dimension of the latent variable space, d, holds significant importance as a hyperparameter in the variational autoencoder training process. After conducting a series of trial tests (with d = 64, 128, 256, 512), a latent size between the encoder and decoder is chosen as 256, which indicates a compression rate of 1/256.

The full dataset is shuffled and split randomly into a training, validation, and test set at a ratio of 0.7:0.15:0.15. During training, the combined reconstruction and KL-divergence loss are minimized using the ADAM optimizer [38] with a learning rate of 1e-5 and a batch size of 8. The first epoch is used to compute the normalization statistics for the scaling of the input/target tensor [50].

Fig. 3b shows a randomly selected example from the test set to give an intuitive understanding of the performance of our VAE model. It visualizes the input distance data to the encoder, while Fig. 3c visualizes the corresponding output distance data from the decoder. Visually, plenty of bright closed-shape regions are distinguishable from a dark background. The reconstructed structure is more blurred than the input one, being a typical issue of the Gaussian kernel. Due to the same reason, some tiny grains are absent. We also calculate the

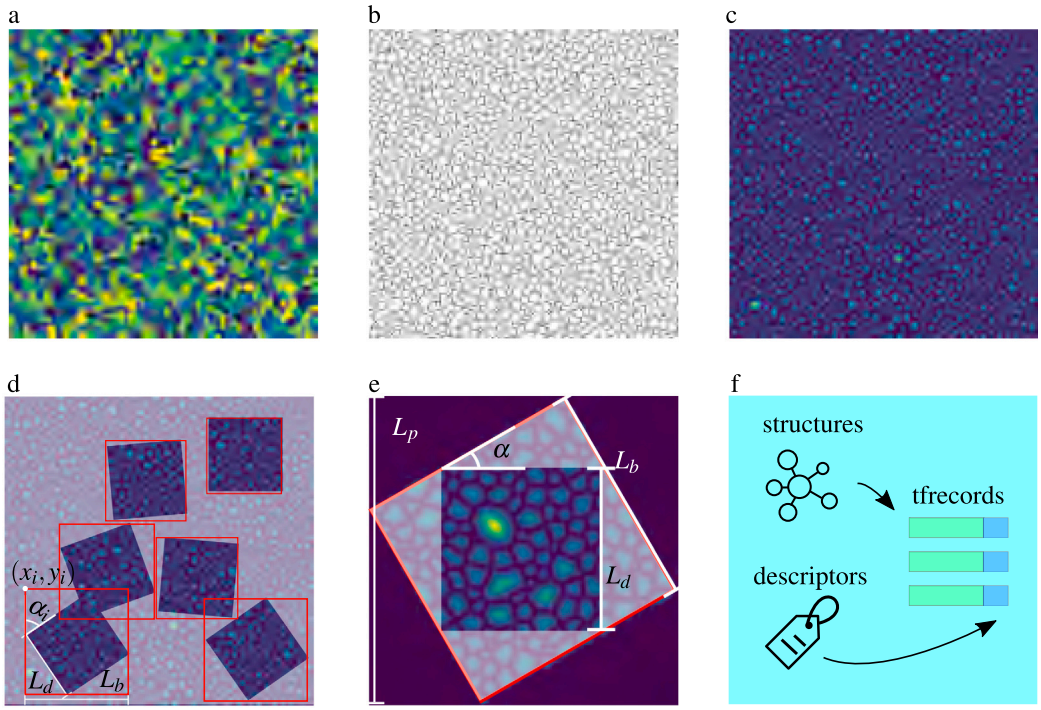

Fig. 2. Pre-processing for computationally generating grain structures data: (a) index data, (b) binary data capturing grain boundaries, (c) distribution of distances of grain bulk centers, (d) illustration of several randomly selected bounding boxes of the region of interest, (e) reorientation of the bounding box, and (f) tfrecords containing the distance data and descriptors.

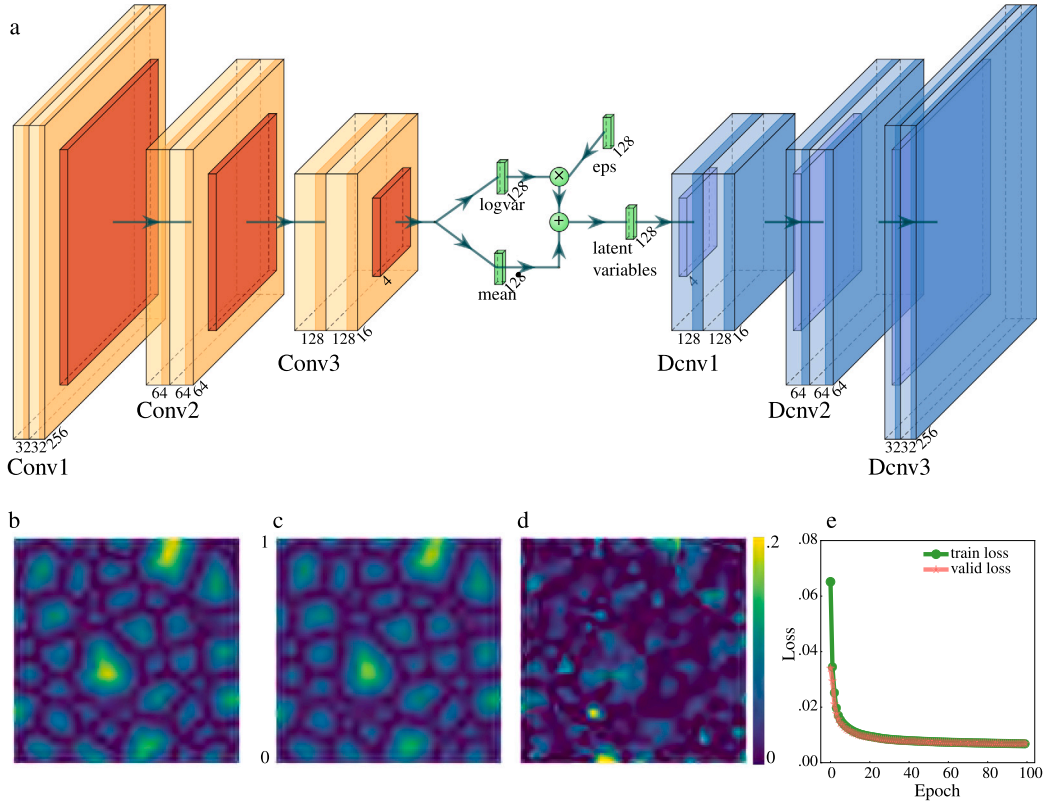

Fig. 3. Data flow in the training process: (a) the VAE consists of the encoder (orange blocks), sampler (green blocks), and decoder (cool blocks), (b) the ground truth from phase-field simulations, (c) the corresponding prediction of the VAE, (d) the residual data, and (e) the loss curves during training epochs.

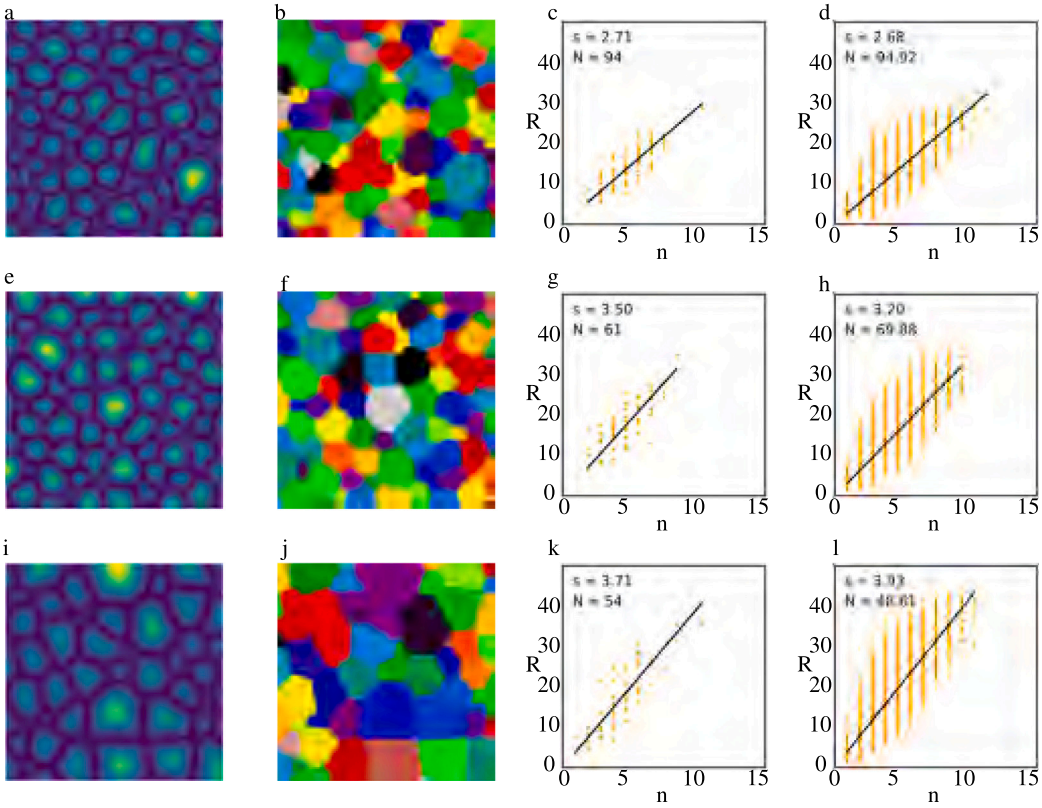

Fig. 4. Characterization of a predicted structure from the test dataset: (a) predicted distance data, (b) segmented index data, (c) dependence of radius (R) and number of neighbors (n) related to the grain structure sample, (d) statistics of 100 randomly selected samples from the early simulation step, (e–h) results of the intermediate simulation step after the same operation, (i–l) results of the late simulation step after the same operation.

difference between the two and plot the residual values in Fig. 3d, where the overall intensity is one order of magnitude lower than the structure itself. This discrepancy is effectively quantified using a widely accepted metric, the mean square error (MSE). As a result, we adopt the MSE as our reconstruction loss, which, together with the KL-divergence loss, constitutes the total loss. In Fig. 3e, we illustrate the total loss of the training dataset (green curve) and the validation dataset (orange curve). Throughout the training epochs, the total loss converges rapidly. Therefore, we postulate that our VAE model can learn the salient feature of grain structure. After training, the losses are 6.80e−3 on the training dataset, 6.94e−3 on the validation dataset. We also evaluate the loss on the test dataset after the training and get a value of 6.78e−3. The similarity in the trends of the loss curves between the test and validation datasets can be attributed to the fact that, in the self-similar state, different grain configurations exhibit statistically similar characteristics. Since training, validation and test error are close, no overfitting could be observed.

3.3. Generation of new sample from decoder

At first, we prove that the distance data can be converted to index data before we move to the generation of new samples. As discussed in the last section, there is a corresponding predicted distance data for each ground-truth input/target distance data. Here we take the prediction distance data for post-processing. Fig. 4 is organized in three rows, where the data are taken from three simulation steps. The later the simulation step, the more grains have grown, shrunk, or even disappeared due to grain growth, which reduces the overall number of grains. For example, Fig. 4a represents the grain structure from the early simulation step. Applying the watershed segmentation method, we obtain the grain patterns in Fig. 4b. A brief explanation of how the watershed algorithm works can be found in the method section. As characteristic quantities of grain structures, the number of grains, the size and face distributions are of interest to material scientists. Such information is contained in Fig. 4c. The radius of grains falls into the range from several pixels to 20 pixels, while the faces are less than 10. We can see a linear correlation between the radius (R) and faces (n). The total number of grains (N) and the slope (s) of the fitted line are marked in the scattering plot. Considering that the decoder of the VAE model generates structures sampled from the latent space, it is helpful to aggregate many generated samples together. We thus concatenate the statistical information of 100 samples together in Fig. 4d. The tendency of a single grain and multiple grains are almost identical, indicating good uniformity. The same data analysis for the intermediate and late simulation steps are shown in Fig. 4e–h and i–l, respectively. An evident tendency concerning the time change is that the slope of fitting curves becomes steeper. Equivalently speaking, the slope is an indicator of the average grain size. For comparison, we can process the ground-truth data from the test set in the same way and visualize it in Fig. S1. Both sets appear to be statistically similar. Nevertheless, two aspects are noticeable. For a single sample, the ground truth's slope is always slightly larger than the prediction. For multiple samples, if it is at the early simulation step, the slope of the prediction is smaller than that of the ground truth, while if it is at the late simulation step, the slope of the prediction is larger than that of the ground truth. This can be attributed to the tiny grains that are not finely reconstructed.

We then investigate the manifold in the latent space to study the performance of the encoder. According to the implementation of the VAE, the encoder provides mean and log-var(iance) vectors that support a normal distribution used to sample latent variables. To verify it, we take all the 4500 samples from the test set to get a statistical distribution in Fig. 5a. The x-axis represents the dimension (Z_n) of

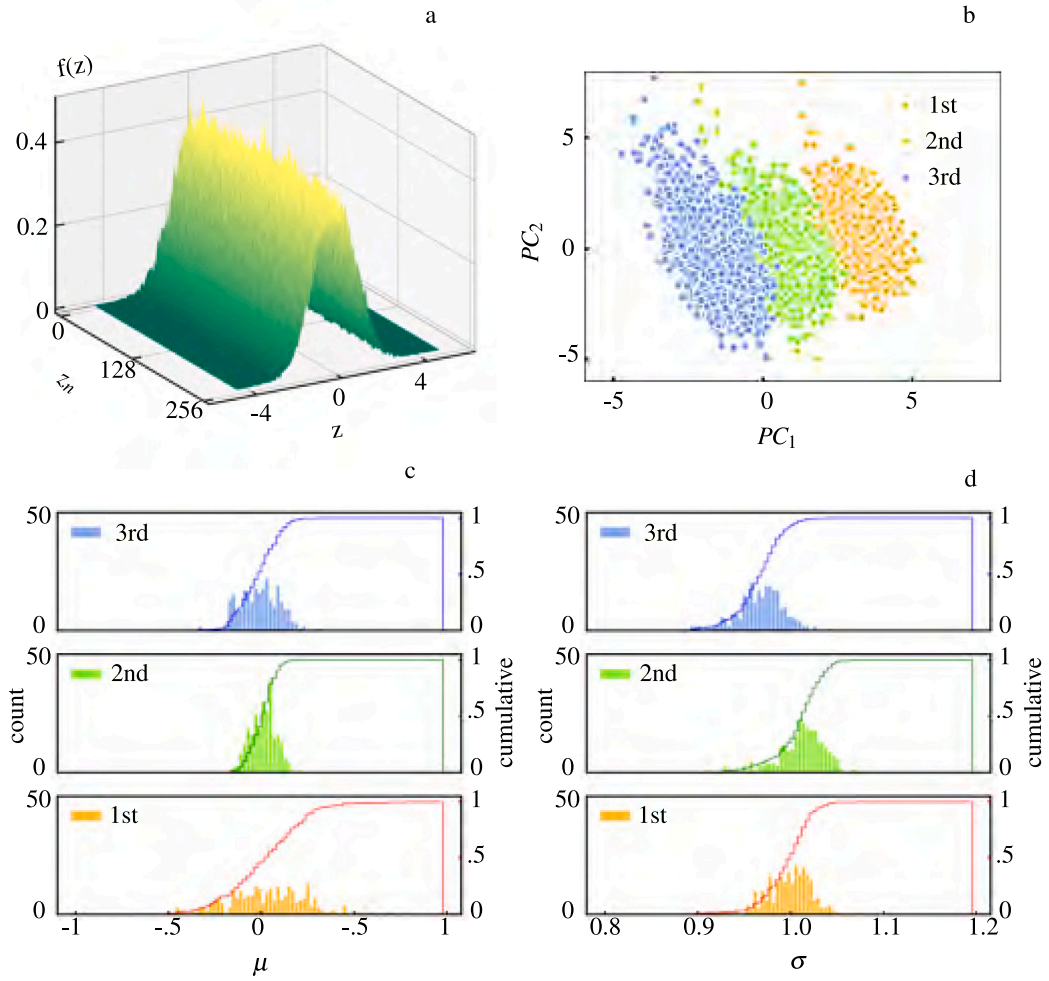

Fig. 5. Analysis of latent space: (a) distribution of latent variables along the latent dimension, (b) projection of latent variables in the principal space with two principle components, PC_1 and PC_2, distribution of (c) mean μ and (d) variance σ for characteristic simulation steps.

the latent vector. The y-axis represents the values (z) of the latent variables. The z-axis represents the probability density ($f(z)$) of the latent variables. We can see that the distribution is approximately invariant with respect to the dimension, indicating that information is evenly distributed in all latent dimensions. In each dimension, the profile of the latent variable is bell-shaped, with a symmetric plane at $y = 0$. Note that we have three simulation steps involved, but no remarkable clustering behavior can be found here. Therefore, we use a principal component analysis [51] to decorrelate our latent variables as much as possible and visualize the results. The effectiveness of PCA has been demonstrated in a series of works, including the compression of the latent space [36], determination of the anisotropy of CT-scanned microstructures [52], and characterization of porous structures [34]. The basic idea of PCA is described in the method section. We compress the 256-dimension variable into 10 components, of which we visualize the two components that explain the most variance. As shown in Fig. 5b, three clusters are formed, corresponding to the simulation steps, which confirms that the neural network can recognize the grain structure. To investigate further, we divide the latent variables in Fig. 5a based on the simulation step descriptors and fit the bell-shaped profile with the normal distributions in Fig. 5c–d. Since we are quantitatively evaluating the profile, therefore inevitably introduce statistical parameters of the latent vector, namely, the mean (μ) and variance (σ) of each latent dimension. These two variables are distinct from the mean vector and log-variance vector shown in Fig. 3a. The curves for mean (μ) and variance (σ) confirm that the fitted parameters of each simulation step have their own characteristics and illustrate that the PCA can detect the difference.

To intuitively understand how the decoder works, we employ some black box tests. We start with an all-zero input to get the “baseline” of the decoder. This results in a chessboard pattern of potential grains with weak contrast, as shown in Fig. 6a. It could be treated as a discretized approximation of an all-zero output. The chessboard pattern appears to be caused by the strided convolution operation. Then we can activate each single dimension one by one. The outputs corresponding to the 1st and the 256th dimensions are displayed in Fig. 6b and c, respectively. The variation can be barely seen. Therefore, we calculate their difference to baseline, as shown in the inset of each figure. We can observe similar phenomena (i) some area is brighter than the background, (ii) the brighter areas are distributed to the whole space, (iii) the brighter area is striped but not closed. Besides, the brighter patterns differ from each other. We believe that such a phenomenon means that each latent variable controls the activation of a unique pattern. Naturally, we infer that the patterns could be combined together. Therefore, we select the number of activated dimensions from an increasing series of values 4, 16, and 64. The obtained outputs are plotted in Fig. 6d–f, respectively. As expected, the more activated dimensions, the more similar the pattern looks like a real grain structure. When there are enough activated dimensions, the chessboard of potential grains in the background is no more visible. With all dimensions activated, in Fig. 6g,

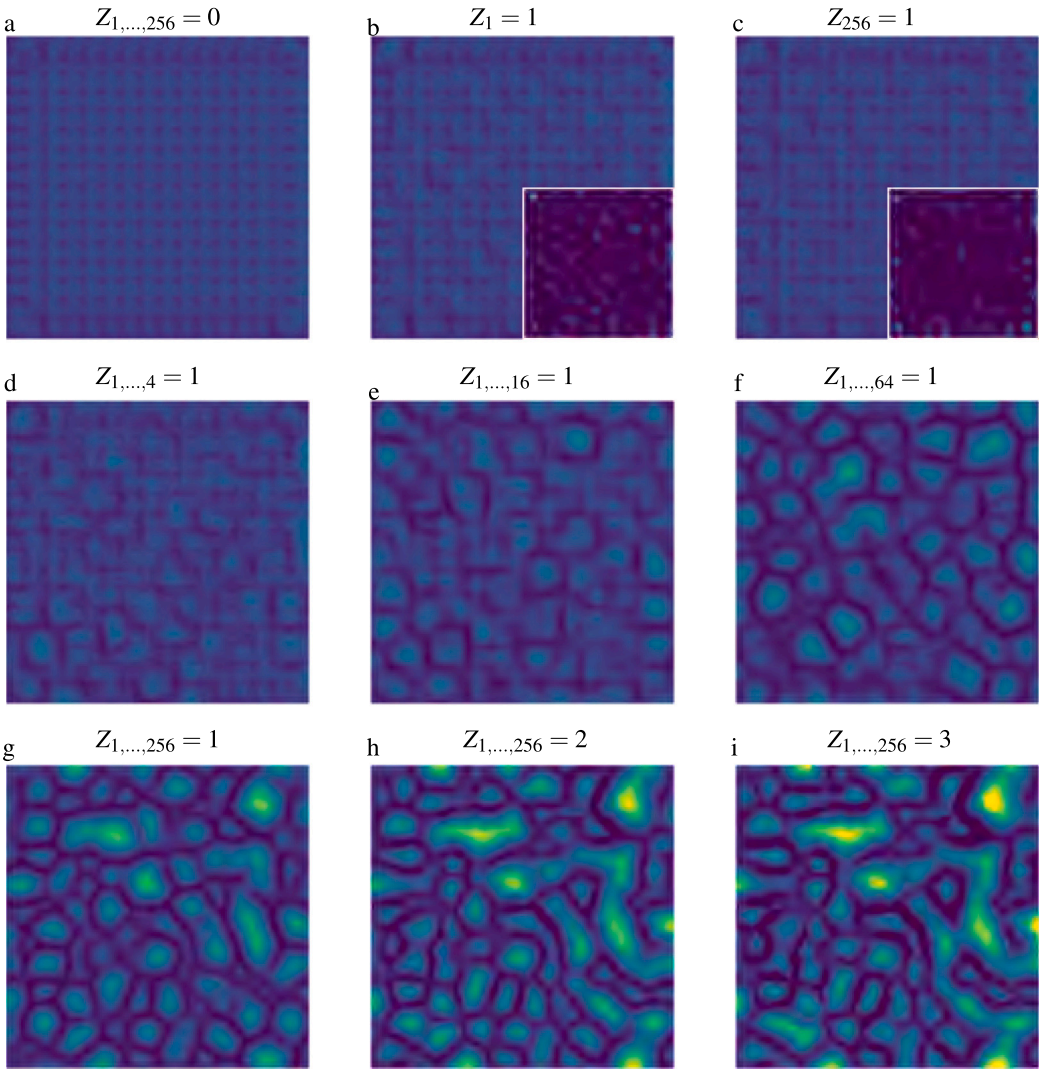

Fig. 6. Decomposition of latent space: (a) baseline, where all the latent variables are deactivated, (b) the decoder approximation of the 1st and (c) of the 256th dimension with positive activation, (d)–(f) the decoder approximation when the number of positively activated dimensions successively increases from 4 to 64, (g)–(l) the decoder approximation when the amplitudes of all latent variables are tested at different levels from 1 to 3, respectively. The inset images in (b) and (c) show their difference to the base line.

we could see more regular bright granular areas. If we continue to level up the magnitude of input values, we find that the intensity increases to saturation while the pattern is fixed. In Figs. 6h and i, the input values are 2 and 3, respectively. The inputs are not restricted to positive values. For the cases where negative inputs are fed to the neural network, similar tendencies can be observed in Fig. S2. To summarize, the decoder resembles a transformation matrix consisting of a series of correlated eigenmodes of grain structure. Correspondingly, the latent variables constitute the eigenvector of the grain structure transformed into the latent space of correlated variables.

Then one may ask how we can combine the eigenmodes to get meaningful grain structures. An arbitrary eigenvector could not produce a meaningful pattern, as seen in the last paragraph. So the idea is to follow the statistical characteristics of the latent variables from the training set. Fig. 5c visualizes the distribution of the mean to explain the clustering behavior. Next, the procedure is repeated, and more tests are carried out. We generate several groups of random vectors from a normal distribution $N(\mu, \sigma)$. In each group, there are 100 samples. The values of the variable μ from −2.0 to 2 with a step of 1.0 and are thus in the vicinity of the values from the training data set. These vectors are transformed by the same PCA model trained in Fig. 5c and the principal components PC_1 and PC_2 of these vectors are determined. Considering PC_1 first, it can be observed that the violin plot's center is monotonically shifting as shown in Fig. 7a. The shifting direction may change subject to the different random initialization in the training process, but its dependence on the mean is definite. Then we vary the variance σ from 1 to 3 with a step of 1.0. After the same projection, in Fig. 7b, we see that the center of the violin plot is constant while the body of the violin plot is elongated. For PC_2, the tendency of mean is the opposite while that of variance is the same, as plotted in Fig. 7c and d, respectively. These results suggest that the PCA is only sensitive to the difference in the mean value of the latent variable in three simulation steps. However, other than the mean value, the order of dimensions matters as well. Therefore, it is impossible to do a reverse PCA to generate meaningful latent variables. We can say that the PCA score is an insufficient but necessary indicator of the grain properties.

We already have 256 pairs of means and variances estimated by fitting all samples into a normal distribution, as done in Fig. 5c–d. They are used to sample vectors for the results in Fig. 8. The arrangement of each row is the same as in Fig. 4. First comes the prediction distance data, followed by its segmentation, then comes the statistics for a single sample, and finally, the aggregated statistics for 100 samples. Because there are three simulation steps, we organize the results into three rows possessing some common features. We can see that the generated

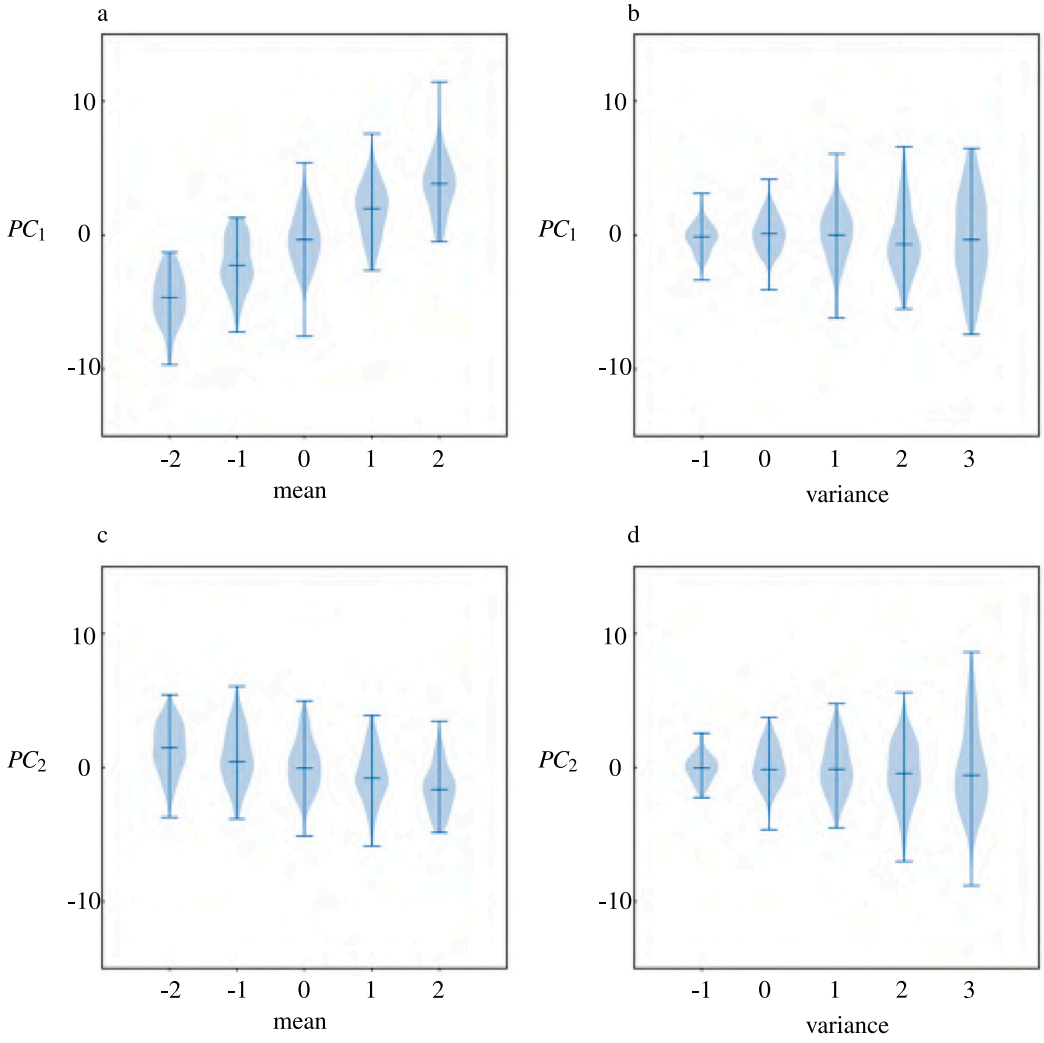

Fig. 7. Mapping a random vector into PCA space: (a) shift of PC_1 towards positive values with respect to the increase of the mean μ, (b) increase of the non-uniformity of PC_1 with respect to the increase of the variance σ, (c) shift of PC_2 towards negative values with respect to the increase of the mean μ, (d) increase of the non-uniformity of PC_2 with respect to the increase of the variance σ.

structures are closer to grain structure than those in Fig. 6g–i, even though more blurry than those in Fig. 4. The segmented grains are neither too round nor too slim. Good linearity is preserved for the statistics of both a single sample and aggregated samples. Furthermore, we observe that the slopes are increasing with respect to time steps. Because there is no ground truth for a single structure to compare, the aggregated statistics is better compared with that from the test dataset. We found that the range of the number of neighbors is less than 10, similar to that in Fig. 4 and Fig. S1. However, the upper bound of size is less than their counterparts, which leads to smaller slopes in all three rows. An explanation is that large grains are rare cases in the training data set, so it is not likely to be predicted by a normal distribution where the occurrence probability of extreme values is low. As a byproduct, the generated samples prefers more uniform grains structures, even though there were no such samples in the training set of the VAE model. This just demonstrates the generative ability of VAE.

4. Conclusion

In this paper, we conduct unsupervised learning of the grain structures, aiming to automatically and massively characterize and generate new samples with similar grain structures. The dataset is obtained from a multi-phase field simulation. Before feeding the data into the model, the Euclidean distance transformation is applied to balance and densify the spatial data. A VAE is trained and can reconstruct the ground-truth structures to realistic performance. We obtained a well-clustered representation in the latent space with the assistance of PCA. Besides, we can generate new grain samples from customized vectors; more importantly, the size distribution can be tuned. In future work, we plan to use the reduced feature to study various structure–property linkages. Moreover, although our demonstration uses grain structure from metal alloys as an example, this method should apply to similar systems, such as bubble foam.

Despite the feasibility, several factors limit our model's performance. Firstly, in the vanilla VAE, information tends to spread in all dimensions, which is unfavorable to compressing the data further. Possible solutions could be the adaption of recently developed VAE derivatives. For example, one can tune the weight of reconstruction loss and KL-divergence loss [53] or force the latent variables into a certain category [54]. Secondly, we notice that the saddle points in the blurry distance data is harmful for segmentation. U-net-inspired architectures could be helpful for this situation [55]. Last, more powerful hardware is desired to accommodate a larger network and more training data.

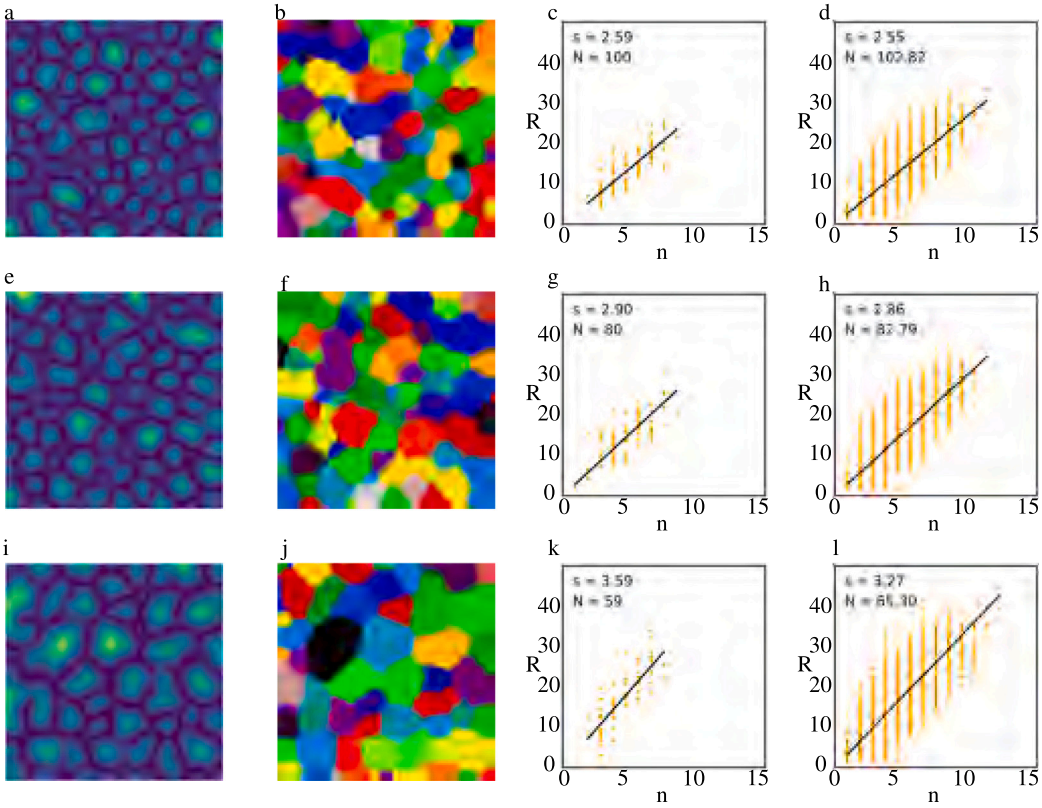

Fig. 8. Characterization of the generated grain structure: (a) generated distance data, (b) segmented index data, (c) dependence of radius (R) and number of neighbors (n) of the corresponding sample, (d) statistics of 100 randomly selected samples from the early simulation step, (e-h) results of the intermediate simulation step after the same operation, (i-l) results of the late simulation step after the same operation.

CRediT authorship contribution statement

Ye Ji: Conceptualization, Methodology, Writing – original draft. **Arnd Koeppe:** Methodology, Writing – original draft, Writing – review & editing, Discussion. **Patrick Altschuh:** Writing – review & editing, Discussion. **Deepalaxmi Rajagopal:** Writing – review & editing, Discussion. **Yinghan Zhao:** Writing – review & editing. **Weijin Chen:** Writing – review & editing. **Yi Zhang:** Writing – review & editing. **Yue Zheng:** Writing – review & editing, Supervision. **Britta Nestler:** Writing – review & editing, Discussion, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The grain data and associated neural network models are stored, managed, and processed within the Kadi4Mat ecosystem [56], publicly accessible through [57]. Reproducibility and data provenance are facilitated by KadiStudio workflows [58] and KadiAI's machine learning workflow concepts [59,60].

Acknowledgments

We thank the Helmholtz Association through the 2020 'Helmholtz-OCPC Program' for the involvement of postdocs in bilateral collaboration projects' for their financial support that enabled this study. This research was supported by the National Natural Science Foundation of China (grant no. 12302430). Further, the authors acknowledge the support by the Federal Ministry of Education and Research (BMBF, Bundesministerium fur Bildung und Forschung) within the FestBatt-Cluster of Competence for Solid-state Batteries (grant nos. 03XP0433D and 03XP0435D), within the ProGraL project (grant no. 03XP0427), and the German Research Foundation (DFG, Deutsche Forschungsgemeinschaft) under Germany's Excellence Strategy through the POLiS Cluster of Excellence – EXC 2154 – Project number 390874152. The authors also acknowledge the support funded by the BMBF and the Ministry of Science, Research and the Arts Baden-Württemberg (MWK-BW) as part of the Excellence Strategy of the German Federal and State Governments in the project ExU-Kadi4X. We thank the BMBF and the MWK-BW for their contributions to the MoMaF-Science Data Center project, which received funding from the state digitization strategy digital@bw (project number 57). Lastly, we would like to acknowledge the Helmholtz Association's support through the program MTET (grant no. 38.02.01) and KNMFI (grant no. 43.31.01).

References

[1] N. Hansen, Hall–Petch relation and boundary strengthening, Scr. Mater. 51 (8) (2004) 801–806, http://dx.doi.org/10.1016/j.scriptamat.2004.06.002, URL https://linkinghub.elsevier.com/retrieve/pii/S1359646204003434.

[2] P.C. Millett, D. Wolf, T. Desai, S. Rokkam, A. El-Azab, Phase-field simulation of thermal conductivity in porous polycrystalline microstructures, J. Appl. Phys. 104 (3) (2008) 033512, http://dx.doi.org/10.1063/1.2964116, URL https://aip.scitation.org/doi/10.1063/1.2964116.

[3] J. Maier, On the conductivity of polycrystalline materials, Berichte der Bunsengesellschaft fuer physikalische Chemie 90 (1) (1986) 26–33, http://dx.doi.org/10.1002/bbpc.19860900105, URL https://onlinelibrary.wiley.com/doi/10.1002/bbpc.19860900105.

[4] J. Von Neumann, Metal Interfaces, 108, American Society for Metals, Cleveland, 1952.

[5] W.W. Mullins, Two-dimensional motion of idealized grain boundaries, J. Appl. Phys. 27 (8) (1956) 900–904, http://dx.doi.org/10.1063/1.1722511, URL https: //aip.scitation.org/doi/10.1063/1.1722511.

[6] M. Hillert, On the theory of normal and abnormal grain growth, Acta Metall. 13 (3) (1965) 227–238, http://dx.doi.org/10.1016/0001-6160(65)90200-2, URL https://www.sciencedirect.com/science/article/pii/0001616065902002.

[7] J.A. Glazier, Grain growth in three dimensions depends on grain topology, Phys. Rev. Lett. 70 (14) (1993) 2170–2173, http://dx.doi.org/10.1103/PhysRevLett.70.2170, URL https://link.aps.org/doi/10.1103/PhysRevLett.70.2170.

[8] R.D. MacPherson, D.J. Srolovitz, The von Neumann relation generalized to coarsening of three-dimensional microstructures, Nature 446 (7139) (2007) 1053–1055, http://dx.doi.org/10.1038/nature05745, URL https://www.nature.com/articles/nature05745.

[9] P.R. Rios, M.E. Glicksman, Polyhedral model for self-similar grain growth, Acta Mater. 56 (5) (2008) 1165–1171, http://dx.doi.org/10.1016/j.actamat.2007.11.010, URL https://www.sciencedirect.com/science/article/pii/S135964540700780X.

[10] B.R. Patterson, D.J. Rule, R.T. DeHoff, V. Tikare, Schlegel description of grain form evolution in grain growth, Acta Mater. 61 (11) (2013) 3986–4000, http://dx.doi.org/10.1016/j.actamat.2013.03.013, URL https://www.sciencedirect.com/science/article/pii/S1359645413002231.

[11] K. Kremeyer, Cellular automata investigations of binary solidification, J. Comput. Phys. 142 (1) (1998) 243–263, http://dx.doi.org/10.1006/jcph.1998.5926, URL https://linkinghub.elsevier.com/retrieve/pii/S0021999198959265.

[12] D. Zöllner, P. Streitenberger, Three-dimensional normal grain growth: Monte Carlo Potts model simulation and analytical mean field theory, Scr. Mater. 54 (9) (2006) 1697–1702, http://dx.doi.org/10.1016/j.scriptamat.2005.12.042, URL https://www.sciencedirect.com/science/article/pii/S135964620600011X.

[13] C.E. Krill III, L.Q. Chen, Computer simulation of 3-D grain growth using a phase-field model, Acta Mater. 50 (12) (2002) 3059–3075, http://dx.doi.org/10.1016/S1359-6454(02)00084-8, URL https://www.sciencedirect.com/science/article/pii/S1359645402000848.

[14] B. Nestler, F. Wendler, M. Selzer, B. Stinner, H. Garcke, Phase-field model for multiphase systems with preserved volume fractions, Phys. Rev. E 78 (1) (2008) 011604, http://dx.doi.org/10.1103/PhysRevE.78.011604, URL https://link.aps.org/doi/10.1103/PhysRevE.78.011604.

[15] D. Fan, L.Q. Chen, Computer simulation of grain growth using a continuum field model, Acta Mater. 45 (2) (1997) 611–622, http://dx.doi.org/10.1016/S1359-6454(96)00200-5, URL https://www.sciencedirect.com/science/article/pii/S1359645496002005.

[16] E. Miyoshi, T. Takaki, M. Ohno, Y. Shibuta, S. Sakane, T. Shimokawabe, T. Aoki, Ultra-large-scale phase-field simulation study of ideal grain growth, npj Comput. Mater. 3 (1) (2017) 1–6, http://dx.doi.org/10.1038/s41524-017-0029-8, URL https://www.nature.com/articles/s41524-017-0029-8.

[17] O. Hunderi, N. Ryum, The kinetics of normal grain growth, J. Mater. Sci. 15 (5) (1980) 1104–1108, http://dx.doi.org/10.1007/BF00551798.

[18] H. Wang, G.Q. Liu, X.Y. Song, J.H. Luan, Topology-dependent description of grain growth, Europhys. Lett. 96 (3) (2011) 38003, http://dx.doi.org/10.1209/0295-5075/96/38003, URL https://iopscience.iop.org/article/10.1209/0295-5075/96/ 38003.

[19] Z. Sun, V. Tikare, B.R. Patterson, A.P. Sprague, Topological characterization of microstructures from 3D digitized voxel data, Comput. Mater. Sci. 55 (2012) 329–336, http://dx.doi.org/10.1016/j.commatsci.2011.11.012, URL https://linkinghub.elsevier.com/retrieve/pii/S0927025611006264.

[20] P.G. Kubendran Amos, R. Perumal, M. Selzer, B. Nestler, Multiphase-field modelling of concurrent grain growth and coarsening in complex multicomponent systems, J. Mater. Sci. Technol. 45 (2020) 215–229, http://dx.doi.org/10.1016/j.jmst.2019.11.011, URL http://www.sciencedirect.com/science/article/pii/S1005030220300608.

[21] A. Bhattacharya, Y.-F. Shen, C.M. Hefferan, S.F. Li, J. Lind, R.M. Suter, C.E. Krill, G.S. Rohrer, Grain boundary velocity and curvature are not correlated in Ni polycrystals, Science 374 (6564) (2021) 189–193, http://dx.doi.org/10.1126/science.abj3210, URL https://www.science.org/doi/10.1126/science.abj3210.

[22] J. Zhang, W. Ludwig, Y. Zhang, H.H.B. Sørensen, D.J. Rowenhorst, A. Yamanaka, P.W. Voorhees, H.F. Poulsen, Grain boundary mobilities in polycrystals, Acta Mater. 191 (2020) 211–220, http://dx.doi.org/10.1016/j.actamat.2020.03.044, URL https://www.sciencedirect.com/science/article/pii/S1359645420302317.

[23] V. Yadav, N. Moelans, Analysis of grain topology and volumetric growth rate relation in three-dimensional normal grain growth, Acta Mater. 156 (2018) 275–286, http://dx.doi.org/10.1016/j.actamat.2018.04.056, URL https://www.sciencedirect.com/science/article/pii/S1359645418303331.

[24] P.R. Rios, D. Zöllner, Critical assessment 30: Grain growth – Unresolved issues, Mater. Sci. Technol. 34 (6) (2018) 629–638, http://dx.doi.org/10.1080/02670836.2018.1434863.

[25] N. Moelans, B. Blanpain, P. Wollants, Quantitative analysis of grain boundary properties in a generalized phase field model for grain growth in anisotropic systems, Phys. Rev. B 78 (2) (2008) 024113, http://dx.doi.org/10.1103/PhysRevB.78.024113, URL https://link.aps.org/doi/10.1103/PhysRevB.78.024113.

[26] J. Carrasquilla, R.G. Melko, Machine learning phases of matter, Nat. Phys. 13 (5) (2017) 431–434, http://dx.doi.org/10.1038/nphys4035, URL http://www.nature.com/doifinder/10.1038/nphys4035.

[27] S. Kalidindi, S.V. Kalinin, T. Lookman, K.K.v. Dam, K. Yager, S. Campbell, R. Farnsworth, M.v. Dam, I. Foster, Handbook On Big Data And Machine Learning In The Physical Sciences, World Scientific, 2020.

[28] S.V. Kalinin, O. Dyck, S. Jesse, M. Ziatdinov, Exploring order parameters and dynamic processes in disordered systems via variational autoencoders, Sci. Adv. 7 (17) (2021) eabd5084, http://dx.doi.org/10.1126/sciadv.abd5084, URL https: //www.science.org/doi/10.1126/sciadv.abd5084.

[29] S.V. Kalinin, B.G. Sumpter, R.K. Archibald, Big–deep–smart data in imaging for guiding materials design, Nature Mater. 14 (10) (2015) 973–980, http://dx.doi.org/10.1038/nmat4395, URL http://www.nature.com/doifinder/10.1038/nmat4395.

[30] S. Karimpouli, P. Tahmasebi, Segmentation of digital rock images using deep convolutional autoencoder networks, Comput. Geosci. 126 (2019) 142–150, http://dx.doi.org/10.1016/j.cageo.2019.02.003, URL https://linkinghub.elsevier.com/retrieve/pii/S0098300418303911.

[31] L. Mosser, O. Dubrule, M.J. Blunt, Stochastic reconstruction of an oolitic limestone by generative adversarial networks, Transp. Porous Media 125 (1) (2018) 81–103, http://dx.doi.org/10.1007/s11242-018-1039-9.

[32] A. Rabbani, M. Babaei, R. Shams, Y.D. Wang, T. Chung, DeePore: A deep learning workflow for rapid and comprehensive characterization of porous materials, Adv. Water Resour. 146 (2020) 103787, http://dx.doi.org/10.1016/j.advwatres.2020.103787, URL https://www.sciencedirect.com/science/article/pii/S0309170820304590.

[33] Y. Zhu, T. Xu, Q. Wei, J. Mai, H. Yang, H. Zhang, T. Shimada, T. Kitamura, T.-Y. Zhang, Linear-superelastic Ti-Nb nanocomposite alloys with ultralow modulus via high-throughput phase-field design and machine learning, npj Comput. Mater. 7 (1) (2021) 1–10, http://dx.doi.org/10.1038/s41524-021-00674-7, Bandiera_abtest: a Cc_license_type: cc_by Cg_type: Nature Research Journals Number: 1 Primary_atype: Research Publisher: Nature Publishing Group Sub-ject_term: Computational methods;Metals and alloys;Structural properties Sub-ject_term_id: computational-methods;metals-and-alloys;structural-properties, URL https://www.nature.com/articles/s41524-021-00674-7.

[34] Y.C. Yabansu, P. Altschuh, J. Hötzer, M. Selzer, B. Nestler, S.R. Kalidindi, A digital workflow for learning the reduced-order structure-property linkages for permeability of porous membranes, Acta Mater. 195 (2020) 668–680, http://dx.doi.org/10.1016/j.actamat.2020.06.003, URL http://www.sciencedirect.com/science/article/pii/S1359645420304274.

[35] P.G. Kubendran Amos, A. Koeppe, R. Perumal, B. Nestler, Data-driven insights on grain-growth kinetics in duplex and triplex systems, SSRN Electron. J. (2022) http://dx.doi.org/10.2139/ssrn.4089125, URL https://www.ssrn.com/abstract=4089125.

[36] D. Montes de Oca Zapiain, J.A. Stewart, R. Dingreville, Accelerating phase-field-based microstructure evolution predictions via surrogate models trained by machine learning methods, npj Comput. Mater. 7 (1) (2021) 3, http://dx.doi.org/10.1038/s41524-020-00471-8, URL http://www.nature.com/articles/s41524-020-00471-8.

[37] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, in: F. Pereira, C. Burges, L. Bottou, K. Weinberger (Eds.), Advances in Neural Information Processing Systems, Vol. 25, Curran Associates, Inc., 2012, pp. 1–9, URL https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[38] D.P. Kingma, M. Welling, Auto-Encoding Variational Bayes, Tech. Rep., 2014, http://dx.doi.org/10.48550/arXiv.1312.6114, URL http://arxiv.org/abs/1312.6114.

[39] J.-j. Shi, W. Zhang, W. Wang, Y.-h. Sun, C.-y. Xu, H.-h. Zhu, Z.-x. Sun, Randomly generating three-dimensional realistic schistous sand particles using deep learning: Variational autoencoder implementation, Eng. Geol. 291 (2021) 106235, http://dx.doi.org/10.1016/j.enggeo.2021.106235, URL https://linkinghub.elsevier.com/retrieve/pii/S0013795221002465.

[40] W. Yan, J. Melville, V. Yadav, L. Yang, K. Everett, M.S. Kesler, A.R. Krause, M.R. Tonks, J.B. Harley, Predicting 2D normal grain growth using a physics-regularized interpretable machine learning model, 2022, arXiv:2203.03735 [physics], URL http://arxiv.org/abs/2203.03735.

[41] Y. Huang, Z. Xiang, M. Qian, Deep-learning-based porous media microstructure quantitative characterization and reconstruction method, Phys. Rev. E 105 (1) (2022) 015308, http://dx.doi.org/10.1103/PhysRevE.105.015308, URL https://link.aps.org/doi/10.1103/PhysRevE.105.015308.

[42] P.G. Kubendran Amos, B. Nestler, Grand-potential based phase-field model for systems with interstitial sites, Sci. Rep. 10 (1) (2020) 22423, http://dx.doi.org/10.1038/s41598-020-79956-x, URL http://www.nature.com/articles/s41598-020-79956-x.

[43] K.G.F. Janssens (Ed.), Computational Materials Engineering: an Introduction to Microstructure Evolution, Elsevier/Academic Press, Amsterdam; Boston, 2007.

[44] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, É. Duchesnay, Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. 12 (85) (2011) 2825–2830, URL http://jmlr.org/papers/v12/pedregosa11a.html.

[45] J. Gostick, M. Aghighi, J. Hinebaugh, T. Tranter, M.A. Hoeh, H. Day, B. Spellacy, M.H. Sharqawy, A. Bazylak, A. Burns, W. Lehnert, A. Putz, OpenPNM: A pore network modeling package, Comput. Sci. Eng. 18 (4) (2016) 60–74, http: //dx.doi.org/10.1109/MCSE.2016.49, Conference Name: Computing in Science Engineering.

[46] J. Hötzer, A. Reiter, H. Hierl, P. Steinmetz, M. Selzer, B. Nestler, The parallel multi-physics phase-field framework Pace3D, J. Comput. Sci. 26 (2018) 1–12, http://dx.doi.org/10.1016/j.jocs.2018.02.011, URL http://www.sciencedirect.com/science/article/pii/S1877750317310116.

[47] H.X. Zhu, S.M. Thorpe, A.H. Windle, The geometrical properties of irregular two-dimensional Voronoi tessellations, Phil. Mag. A 81 (12) (2001) 2765–2783, http: //dx.doi.org/10.1080/01418610010032364, URL http://www.tandfonline.com/doi/abs/10.1080/01418610010032364.

[48] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous distributed systems, 2016, Number: arXiv:1603.04467, [cs], URL http://arxiv.org/abs/1603.04467.

[49] A. Koeppe, Deep Learning in the Finite Element Method, Vol. Rheinisch-Westfälische Technische Hochschule Aachen, RWTH Aachen University, 2021, http://dx.doi.org/10.18154/RWTH-2021-04990, Artwork Size: pages 1 Online-Ressource: Illustrationen, Diagramme Pages: pages 1 Online-Ressource: Illustrationen, Diagramme Publication Title: Dissertation, URL https://publications.rwth-aachen.de/record/819355.

[50] A. Koeppe, F. Bamer, B. Markert, An efficient Monte Carlo strategy for elasto-plastic structures based on recurrent neural networks, Acta Mech. 230 (9) (2019) 3279–3293, http://dx.doi.org/10.1007/s00707-019-02436-5.

[51] I.T. Jolliffe, Principal component analysis and factor analysis, in: I.T. Jolliffe (Ed.), Principal Component Analysis, in: Springer Series in Statistics, Springer, New York, NY, 1986, pp. 115–128, http://dx.doi.org/10.1007/978-1-4757-1904-8_7.

[52] R.A. Salo, I. Belevich, E. Manninen, E. Jokitalo, O. Gröhn, A. Sierra, Quantification of anisotropy and orientation in 3D electron microscopy and diffusion tensor imaging in injured rat brain, NeuroImage 172 (2018) 404–414, http://dx.doi.org/10.1016/j.neuroimage.2018.01.087, URL https://www.sciencedirect.com/science/article/pii/S1053811918300879.

[53] I. Higgins, L. Matthey, A. Pal, C.P. Burgess, X. Glorot, M. Botvinick, S. Mohamed, A. Lerchner, Beta-VAE: Learning basic visual concepts with a constrained variational framework, in: ICLR, 2017, pp. 1–22.

[54] A.v.d. Oord, O. Vinyals, K. Kavukcuoglu, Neural discrete representation learning, 2018, http://dx.doi.org/10.48550/arXiv.1711.00937, Number: arXiv:1711.00937, [cs] version: 2, URL http://arxiv.org/abs/1711.00937.

[55] O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional networks for biomedical image segmentation, in: N. Navab, J. Hornegger, W.M. Wells, A.F. Frangi (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, in: Lecture Notes in Computer Science, Springer International Publishing, Cham, 2015, pp. 234–241, http://dx.doi.org/10.1007/978-3-319-24574-4_28.

[56] N. Brandt, L. Griem, C. Herrmann, E. Schoof, G. Tosato, Y. Zhao, P. Zschumme, M. Selzer, Kadi4Mat: A research data infrastructure for materials science, Data Sci. J. 20 (1) (2021) 8, http://dx.doi.org/10.5334/dsj-2021-008, URL http: //datascience.codata.org/articles/10.5334/dsj-2021-008/.

[57] Y. Ji, Dataset of "towards automatic feature extraction and sample generation of grain structure by variational autoencoder", 2022, http://dx.doi.org/10.5281/zenodo.7463217.

[58] L. Griem, P. Zschumme, M. Laqua, N. Brandt, E. Schoof, P. Altschuh, M. Selzer, KadiStudio: FAIR modelling of scientific research processes, Data Sci. J. 21 (1) (2022) 16, http://dx.doi.org/10.5334/dsj-2022-016, URL http://datascience.codata.org/articles/10.5334/dsj-2022-016/.

[59] A. Koeppe, F. Bamer, M. Selzer, B. Nestler, B. Markert, Explainable artificial intelligence for mechanics: Physics-explaining neural networks for constitutive models, Front. Mater. 8 (2022) URL https://www.frontiersin.org/article/10.3389/fmats.2021.824958.

[60] A. Koeppe, F. Bamer, M. Selzer, B. Nestler, B. Markert, Workflow concepts to model nonlinear mechanics with computational intelligence, PAMM 21 (1) (2021) e202100238, http://dx.doi.org/10.1002/pamm.202100238, URL https: //onlinelibrary.wiley.com/doi/abs/10.1002/pamm.202100238, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/pamm.202100238.