Database development and exploration of process–microstructure relationships using variational autoencoders

Srihari Sundar, Veera Sundararaghavan*

Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48105, USA

ARTICLE INFO

ABSTRACT

The paper demonstrates graphical representation of a large database containing process–microstructure relationships using an unsupervised machine learning algorithm. Correlating microstructural features to processing is an essential first step to answer the difficult problem of process sequence design. In this paper, a large database of 346,200 orientation distribution functions resulting from a variety of process sequences is constructed, where each sequence comprises up to four stages of tension, compression and rolling along different directions in various permutations. This open-source database is constructed for collaborative development of process design algorithms. The paper demonstrates a novel application of the large database: graphical representation of texture–process relationships. A variational autoencoder is used to reduce the entire database to a two dimensional latent space where variations in processes and properties can be visualized. Using proximity analysis in this latent space, we can quickly unearth multiple process solutions to the problem of texture or property design.

1. Introduction

Properties of metallic alloys such as elastic modulus, strength, and thermal conductivity are dependent on the substructure of the material at lengths of few hundred microns and below. This microstructure is composed of an aggregate of micro-scale grains (or crystals) of various shapes and sizes. Each grain is distinguished by its crystallographic orientation, which quantifies spatial orientation of the atomic lattice. Descriptors such as the probability density function for orientations: the orientation distribution function (ODF) describes the texture or the distribution of orientations in the microstructure and is an important determiner of properties in polycrystalline alloys. Given the texture, engineering properties can be computed through averaging relationships. The inverse problem of identifying microstructural features that lead to a given set of properties has been addressed in the past by several authors using optimization techniques [1,2] or data-mining methods [3–8]. In the case of identifying textures that lead to a known set of properties, typically linear programming or gradient optimization methods [1,2,6] have been employed, while in recent times, data mining methods has been used for increased efficiency [9]. While these methods can identify optimal microstructures, the problem of designing processes to achieve optimal microstructures has comparatively seen less attention.

Microstructures can be tailored so that desired properties can be achieved through controlled deformation or thermal treatment. Recent work has focused on identifying process parameters, for example, scanning width in additive manufacturing [10] or the initial texture that can be cold rolled to achieve a desired final texture [11], but the problem of sequence identification is largely unexplored. A typical process for manufacturing a component such as a turbine blade can contain as many as twenty distinct processing stages, each taking the microstructure from one state to another, eventually achieving a desired microstructure. The problem of "processing path design" aims to discover a sequence of known processes to realize microstructures with optimal properties [12–19]. In [15], a model capable of predicting texture evolution starting from any specific deformation path (as processing path lines) was developed to guide process design. Because of non-uniqueness in processing path solution (different processing paths leading to similar microstructural features) and complex nature of the microstructure–property-process relationships, the problem of identifying an optimal process pathline is better solved using data-mining strategies [11,17,20,21]. In [12,13], this problem was addressed by using proper orthogonal decomposition to project textures obtained from different processes in a reduced order 'process plane' where the process sequence needed to obtain a given ODF was obtained via linear programming. In [17], a large database of process lines was developed

---

* Corresponding author.

E-mail address: veeras@umich.edu (V. Sundararaghavan).
URL: http://umich.edu/-veeras (V. Sundararaghavan).

and superimposed in a property closure to identify optimal process sequences using graph search techniques. While these techniques have allowed exploration of innovative process design techniques, they do not yet use the advantages offered by modern machine learning methods for achieving significant model reduction directly in the texture space. For example, Ref. [17] indicates the need for a 12-dimensional subspace to adequately capture the texture dependence of the macroscale plastic properties of interest in fcc metals.

In recent years, neural network based generative models have become popular and are capable of generating a continuous low dimensional space of input data that can be graphically visualized. In this paper, a Variational Autoencoder (VAE) network is used to achieve a continuous latent space for the entire database. Using proximity analysis in this latent space, multiple process solutions to the problem of texture or property design is demonstrated. To train this network, we focus on the development of a computationally generated open source database of textures of single phase FCC metals that result from sequences of up to four stages of tension, compression and rolling along different directions in various permutations. Such a database can be a platform for collaborative testing and assessing machine learning algorithms by computer scientists who do not have domain knowledge in texture evolution. Visualization of these large databases has been a challenge. In the past, reduced methods such as proper orthogonal decomposition have been used to reduce the texture space [20,21], but they have usually resulted in large dimensionality of reduced spaces. One task that is accomplished in this work is the direct visualization of this database containing 346200 textures in two dimensions using the autoencoding technique.

2. Methodology

2.1. Process model used in the database

Imposed deformation is described by the deviatoric[1] velocity gradient tensor, **L**. The velocity gradient tensor itself can be expressed as a linear combination of multiple processes, as shown in the equation below. The first term in the equation represents tension in the X direction, second term represents rolling along Y (with Z being the short transverse direction), the next three terms represent pure shear along the three directions and the last three terms represent rotations about the three axes. A linear combination of these terms can be used to describe all possible volume preserving deformations. Tension, compression and rolling being the deformation modes employed in this study, the $\alpha_i's$ used to describe these processes are shown in Table 1.

Besides the 6 processes, denoted by P_i with $i = [1, 6]$ referring to Table 1, strain rate is another parameter which can be used, and this is denoted by β. This will control the amount of effective deformation at the end of a time step for any particular process. The set of β values chosen in this work are $\beta_i = \{-1, -0.5, 0.5, 1\}$. The rolling ZY notation, with a positive β denotes that the rolling direction is Y and the compression direction is Z, with no strain in the transverse X direction. On the other hand, rolling ZY notation, with a negative β denotes the opposite, i.e., rolling direction is Z and the compression direction is Y.

2.2. Material model and ODF evolution

We employ the axis-angle parametrization of the orientation space proposed by Rodrigues [20] as $\mathbf{r} = \mathbf{n} \tan(\frac{\theta}{2})$. Given the Rodrigues vector $\mathbf{r}$, the rotation tensor $\mathbf{R}$ transforming quantities from the crystal frame to the sample frame is takes the form

$$\mathbf{R} = \frac{1}{1 + \mathbf{r}.\mathbf{r}}(\mathbf{I}(1 - \mathbf{r}.\mathbf{r}) + 2(\mathbf{r} \otimes \mathbf{r} + \mathbf{I} \times \mathbf{r})). \tag{1}$$

---

[1] Since plastic deformation via dislocation slip is isochoric.

Table 1
Coefficients for describing tension and Rolling in different planes.

$$\mathbf{L} = \alpha_1 \begin{bmatrix} 1 & 0 & 0 \\ 0 & -0.5 & 0 \\ 0 & 0 & -0.5 \end{bmatrix} + \alpha_2 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

$$+ \alpha_3 \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \alpha_4 \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$+ \alpha_5 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} + \alpha_6 \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$+ \alpha_7 \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} + \alpha_8 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

Process	α_1, α_2
Tension X	1,0
Tension Y	$-0.5, 0.75$
Tension Z	$-0.5, -0.75$
Rolling YX	$1, -0.5$
Rolling ZY	$0, 1$
Rolling XZ	$-1, -0.5$

The database contains the fundamental region of the orientation space in which each crystal orientation is represented uniquely. The fundamental region for the cubic symmetry group as used in this paper results in a truncated cube. The ODF value (represented by $\mathcal{A}(\mathbf{r}, t)$) defined on the fundamental region describes the probability density corresponding to orientation $\mathbf{r}$ at time t.

The evolution of ODF is governed by the ODF conservation equation similar to the continuity equation in continuum mechanics, which expressed in Eulerian form is given by [22]:

$$\frac{\partial \mathcal{A}(\mathbf{r}, t)}{\partial t} + \nabla \mathcal{A}(\mathbf{r}, t)\cdot\mathbf{v}(\mathbf{r}, t) + \mathcal{A}(\mathbf{r}, t)\nabla\cdot\mathbf{v}(\mathbf{r}, t) = 0 \tag{2}$$

where $\mathbf{v}(\mathbf{r}, t)$ is the Eulerian reorientation velocity.

From Eq. (2), it is seen that the evolution of the ODF is controlled by the reorientation velocity $\mathbf{v}(\mathbf{r}, t)$. The velocity gradient for the process, $\mathbf{L}$ is linked to $\mathbf{v}(\mathbf{r}, t)$. The reorientation velocity is evaluated through crystal constitutive relations, which involve the crystal velocity gradient. The velocity gradient of a crystal with orientation, $\mathbf{r}$, yields the following form:

$$\mathbf{L} = \mathbf{\Omega} + \mathbf{R} \sum_{\alpha} \dot{\gamma}^{\alpha} \bar{\mathbf{T}}^{\alpha} \mathbf{R}^T \tag{3}$$

where Ω is the lattice spin, $\dot{\gamma}^{\alpha}$ is the shearing rate along the slip system α and $\bar{\mathbf{T}}^{\alpha}$ is the Schmid tensor for the slip system α, given by $(\bar{\mathbf{m}}^{\alpha} \otimes \bar{\mathbf{n}}^{\alpha})$, where $\bar{\mathbf{m}}^{\alpha}$ is the slip direction and $\bar{\mathbf{n}}^{\alpha}$ is the slip plane normal, both in the crystal lattice frame. The expressions for the spin and symmetric parts are obtained as shown below:

$$\mathbf{\Omega} = \mathbf{W} - \sum_{\alpha} \dot{\gamma}^{\alpha} \mathbf{R} \bar{\mathbf{Q}}^{\alpha} \mathbf{R}^T \tag{4}$$

$$\bar{\mathbf{D}} = \sum_{\alpha} \dot{\gamma}^{\alpha} \bar{\mathbf{P}}^{\alpha} \tag{5}$$

where $\bar{\mathbf{P}}^{\alpha}$ and $\bar{\mathbf{Q}}^{\alpha}$ are the symmetric and skew parts of the Schmid tensor respectively and $\bar{\mathbf{D}}$ is the macroscopic deformation rate expressed in the lattice frame through, $\bar{\mathbf{D}} = \mathbf{R}^T \mathbf{D} \mathbf{R}$. The shearing rate on slip systems is given by a power law and we further assume that all slip systems have identical hardness.

$$\dot{\gamma}^{\alpha} = \dot{\gamma}^0 \left(\frac{\tau^{\alpha}}{s}\right)^{\frac{1}{m}} \text{sgn}\left((\frac{\tau^{\alpha}}{s})\right) \tag{6}$$

where s is the slip system hardness, m is the strain rate sensitivity, $\dot{\gamma}^0$ is a reference rate of shearing and τ^{α} is the resolved shear stress on slip system α. Further, the resolved stress is related to the crystal Cauchy stress as
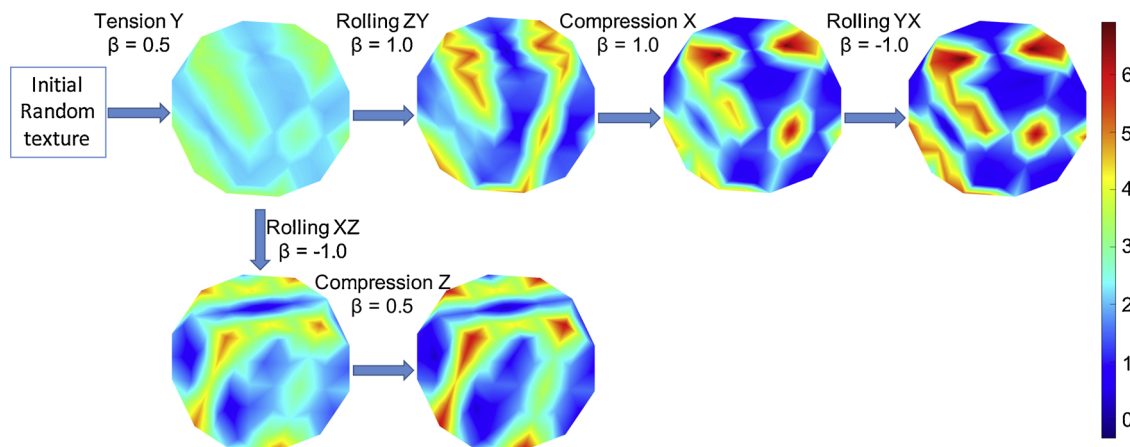
Fig. 1. ODF evolution starting from an initial random texture through two process sequences. Both sequences share the same first stage, with the top depicting a four stage sequence and the bottom row depicting three stage sequence.

$$\tau^\alpha \;=\; \bar{\boldsymbol{\sigma}} \cdot \bar{\boldsymbol{P}}^\alpha \tag{7}$$

By solving the system of Eqs. (5)–(7), the crystal cauchy stress ($\bar{\sigma}$) and the shear rate ($\dot{\gamma}^\alpha$) can be evaluated. Next, using Eq. (4), we can evaluate the lattice spin vector as,

$$\omega \;=\; \mathrm{vect}(\boldsymbol{\Omega}) \tag{8}$$

which is then used to evaluate the reorientation velocity as,

$$\mathbf{v} \;=\; \frac{1}{2}(\omega \;+\; (\omega \cdot \mathbf{r})\mathbf{r} \;+\; \omega \otimes \mathbf{r}) \tag{9}$$

Finally, the ODF, $\mathcal{A}$, over the current fundamental region $\mathcal{R}$ is evaluated from the Eulerian form (Eq. (2)) of the conservation equation. Eq. (2) is solved using a finite element formulation developed in Ref. [22].

2.3. Database structure

With a combination of four strain rates and six possible processes, there are a total of 24 processes that can be used in all permutations in each stage of a process sequence to make up the process sequence-ODF database. In the current work sequences containing up to 4 stages have been considered. An example of what the transformation of ODF after each stage in a sequence of three stages can be seen in Fig. 1. This gave a total of 346,200 sequence-ODF pairs and this was split into 250,000 data points for the training and a little less than 50,000 data points for the validation and test sets.

2.4. Variational auto encoder

An auto encoder is a learning model which consist of an encoder and a decoder neural network. The encoder takes the input data from a high dimensional space and maps it to a code which is generally of lower dimension. The decoder takes this lower dimensional code and maps it back to the original space. Training of the model is performed by reducing the error between what is input to the encoder and what is output by the decoder. While a one-one mapping is a possible outcome, this is avoided by having a code dimension smaller than the input dimension. The objective is to let the model uncover useful properties in the input data while the decoder output only serves the purpose of finding the encodings. When the encoder and decoder networks are single layer perceptron with linear activation functions and a mean squared loss function, the model is reduced to a principal component analysis (PCA) [23,24].

A variational auto encoder (VAE) is a class of auto encoders where the latent representation is space spanning, and can be used as a generative model [25,26]. Fig. 2 shows a schematic of a VAE. The model is trained by minimizing both the reconstruction error and a Kullback–Leibler (KL) divergence. This divergence measures how the mean and standard deviation from the encoder diverges from the target distribution from which the input data is hypothesised to be generated. A latent space thus "learnt" can be used in interpolatory and extrapolatory modes and hence the model's generative capabilities. VAEs have been employed in many domains: chemical design by generating new drug-like molecules and optimizing for properties [27], designing new materials possessing desired optical absorption properties [28], and discovering hypothetical metastable vanadium oxides [29].

Thus, VAEs give a reduced order representation of the data set which can be visualized on low dimensional plots and any point can be sampled from the low dimensional space to produce an output in the original space. This makes VAE attractive to use on the database described in Section 2.3, for dimensionality reduction, reconstruction and proximity analysis based on common features.

For each ODF input to the model, a vector containing the latent space coordinates (same as the mean in Fig. 2) is obtained when parsed through the encoder. In the case of a 2D latent space this can be visualized on a 2D scatter plot and Fig. 3 shows the ODFs from the training set mapped onto the latent space. The points are colored by Young's modulus along the X direction in Fig. 3a, and it is seen that there are certain regions that contain hot spots. Fig. 3b shows the points colored by the last process from the sequence which resulted in that particular texture. The supplementary video shows how specific data points are clustered together based on the last process of the sequence.

The model being generative, points can be sampled from a grid on the latent space and parsed through the decoder, to understand the spatial dependence of ODF mapped on the latent space. Fig. 4 shows this sampling. It is seen that the corners represent different families of material textures, and farther away from origin, higher the intensity at particular nodes in the ODF. Different regions in the latent space are seen to encode specific features in the ODF space, and this feature separation will be useful for analysis of performance as shown in the next section.

3. Results and discussion

One hyperparameter that needs to be identified is optimal dimension of the latent space. In this vein, multiple VAEs were trained with the latent space ranging between 1 and 5 dimensions. One of the main criteria for a "well trained" model for the current application would be the optimal identification of important features from the dataset. For this purpose, the quantitative estimate through distribution of reconstruction error and qualitative analysis by way of capturing important ODF features are carried from the different trained VAEs. The
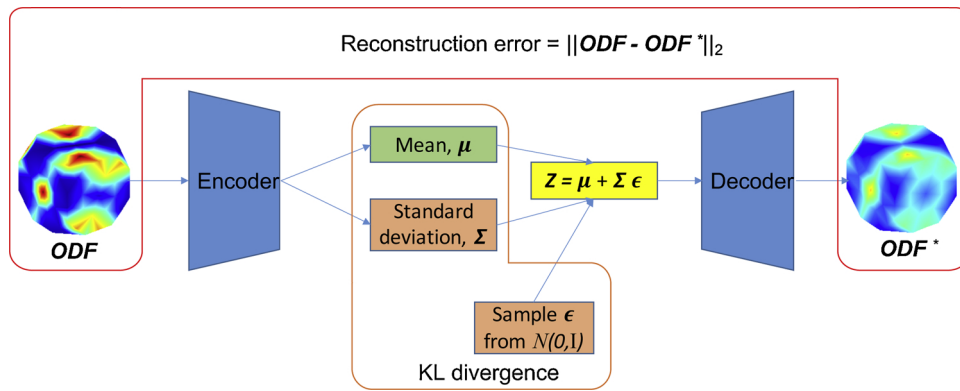
Fig. 2. Schematic of a VAE showing reconstruction of an ODF and error measures used for training the model.
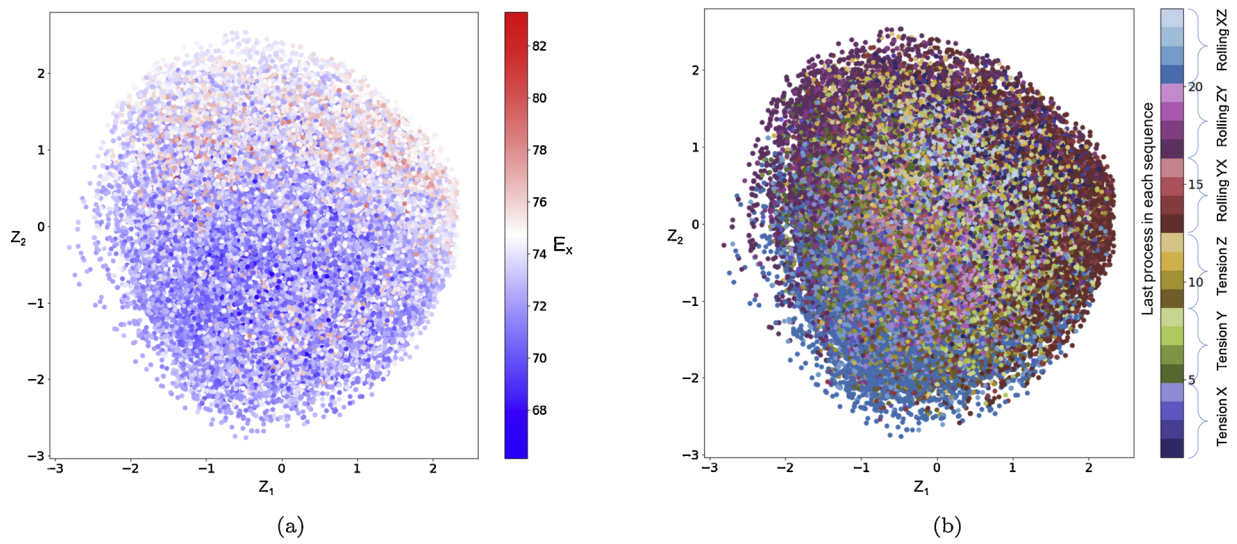

(a)

(b)

Fig. 3. Each point in the latent space representation corresponds to one ODF from the training set. Figure (a) shows points colored by E_x and figure (b) shows points colored by the last process from each sequence.
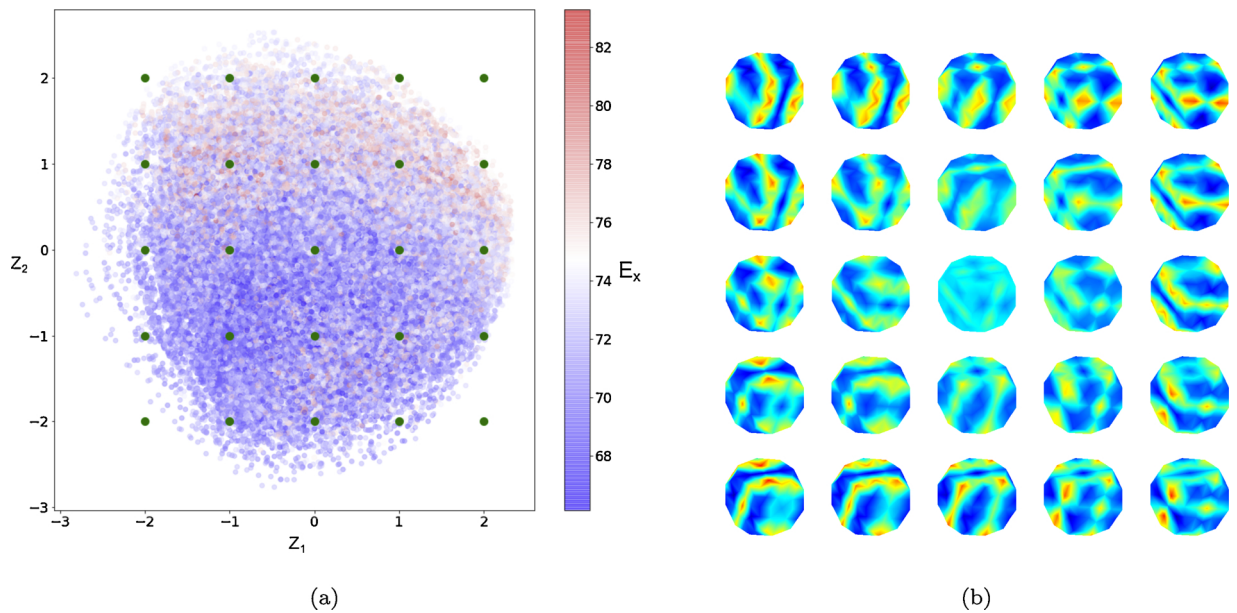

(a)

(b)

Fig. 4. This figure illustrates the generative capabilities of the trained model, including extrapolatory reconstruction. Figure (a) depicts the points from the 2D latent space that were sampled for reconstruction and figure (b) shows reconstructed ODFs.
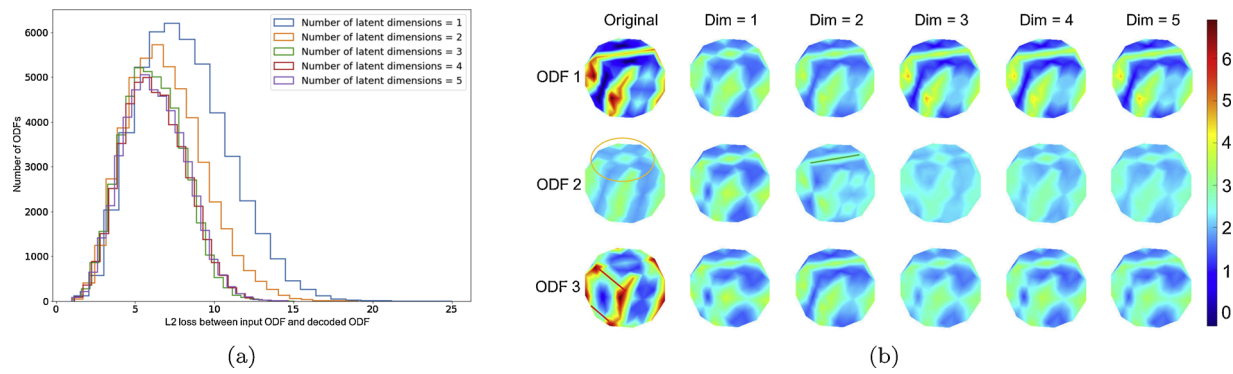
(a)

(b)

Fig. 5. Quantitative and qualitative study of VAE performance to understand optimal number of dimensions in the latent space. Figure (a) shows different histograms of L2 error between ODFs in the validation dataset and their reconstruction for varying latent dimensions. Figure (b) shows 3 ODFs to illustrate preservation of important features, marked with fiber lines, by the encoder and subsequent reconstruction.
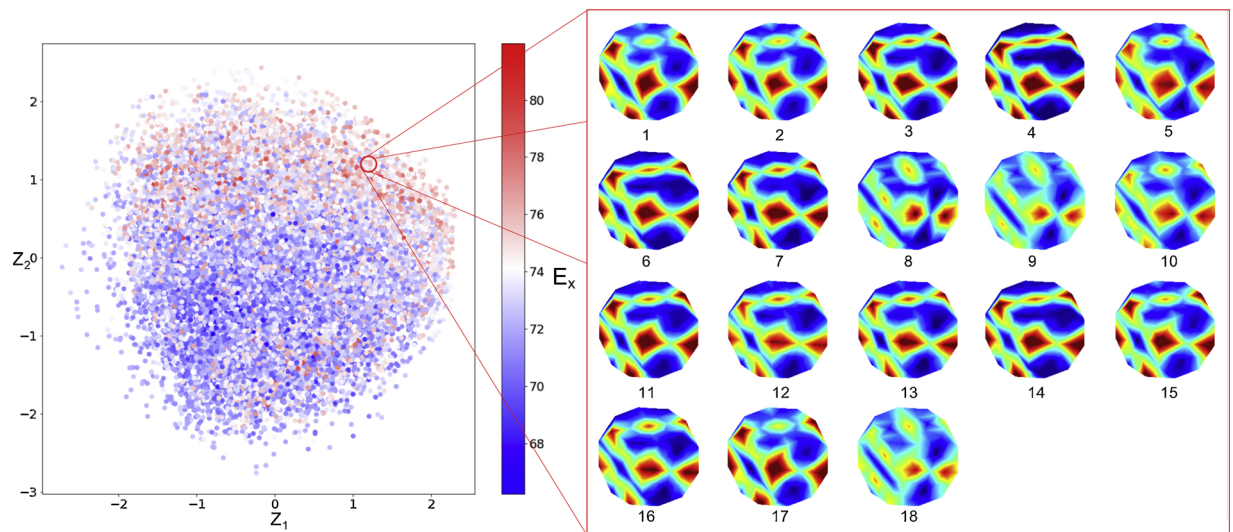

Fig. 6. ODFs from the validation set encoded onto latent space. The red circle marks a circular area with radius 0.1 units around the point (1.2, 1.2) in the latent space and ODFs encoded in that region are shown on the right.

Table 2
Process sequences leading to ODFs in the red area from Fig. 6.

	Stage 1	β_1	Stage 2	β_2	Stage 3	β_3	Stage 4	β_4	E_x
1	Rolling YX	−1.0	Tension Z	1.0	Rolling YX	−0.5	Tension Z	0.5	73.197
2	Tension Z	0.5	Compression X	0.5	Tension Z	1.0	Rolling YX	−1.0	73.977
3	Tension Z	0.5	Tension Z	1.0	Rolling YX	−1.0	Compression X	1.0	75.368
4	Compression X	1.0	Rolling YX	−1.0	Compression X	11.0	Tension Z	1.0	76.811
5	Rolling XZ	1.0	Compression Y	1.0	Rolling YX	−1.0	Rolling ZY	−1.0	73.295
6	Compression X	1.0	Tension X	0.5	Rolling YX	−1.0	Tension Z	1.0	75.908
7	Rolling YX	−1.0	Rolling ZY	−1.0	Rolling YX	−1.0	Rolling XZ	1.0	75.543
8	Compression X	1.0	Rolling YX	−1.0	Rolling XZ	−0.5	Rolling ZY	−1.0	74.941
9	Rolling ZY	−0.5	Rolling ZY	−0.5	Rolling YX	−0.5	Rolling YX	−0.5	73.925
10	Compression X	0.5	Tension Z	0.5	Rolling YX	−1.0	Rolling ZY	−0.5	74.404
11	Tension Z	1.0	Compression X	1.0	Rolling YX	−1.0	Tension Z	0.5	74.956
12	Compression Y	0.5	Tension Z	1.0	Compression X	1.0	Rolling YX	−1.0	76.001
13	Rolling ZY	−1.0	Rolling YX	−1.0	Rolling XZ	1.0	Rolling YX	−0.5	75.259
14	Compression X	1.0	Rolling YX	−1.0	Compression X	0.5	Tension Z	1.0	75.909
15	Tension Z	0.5	Tension Z	1.0	Rolling YX	−1.0	Compression X	0.5	74.108
16	Rolling YX	−1.0	Rolling ZY	−0.5	Compression X	1.0	Tension Z	0.5	75.400
17	Tension Z	0.5	Rolling YX	−0.5	Rolling YX	−1.0	Tension Z	1.0	73.229
18	Compression Z	0.5	Rolling ZY	−1.0	Tension Z	0.5	Rolling YX	−1.0	74.395

validation dataset is used for this purpose. Fig. 5a captures the variation of reconstruction error histogram, vector norm of the difference between input ODF and its reconstruction, with dimension of the latent space. It is seen that there is a loss in intensity at many nodes in the ODF. This is attributed to the scaling of input data, and descaling of the output. This is required to keep all attributes of the input data between 0 and 1, which is one of the assumptions of the methods used, and the NN layers.

Since capturing useful features of data into the latent space is more important than exact reconstruction, Fig. 5b shows a study of model
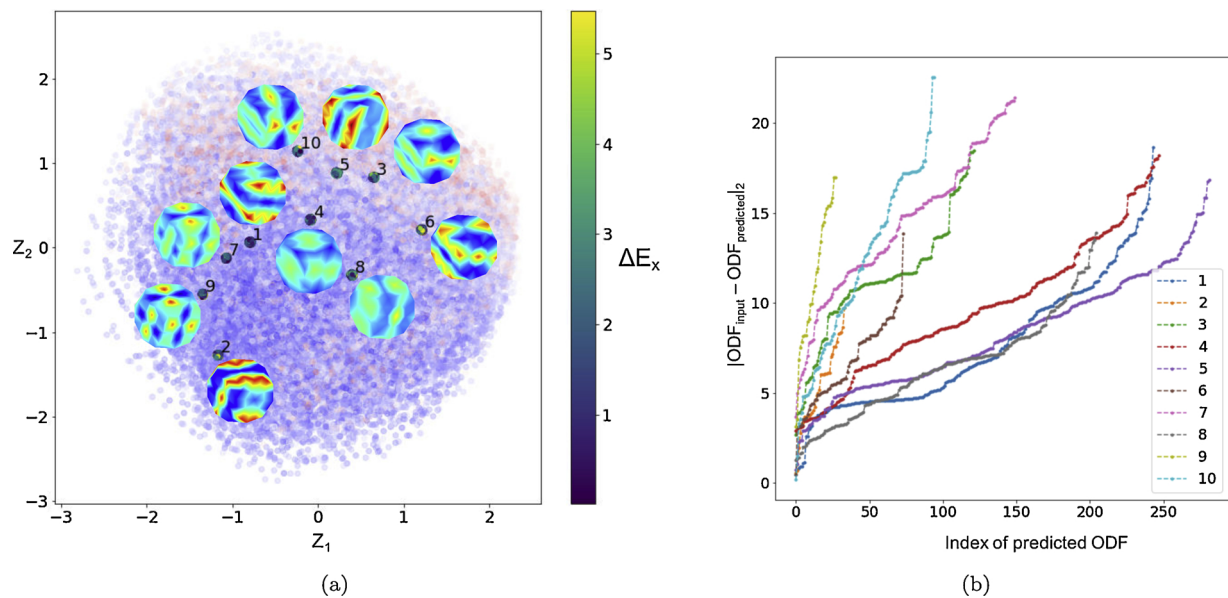
Fig. 7. Figure (a) shows a few ODFs used for sequence identification from the test set. The encodings from these are overlayered on top of encodings from training set, and points from the training set in a small area around the test encodings (predicted ODFs) are colored by the difference in E_x from test ODFs. Figure (b) shows the L2 difference between input and predicted ODFs for each test case.
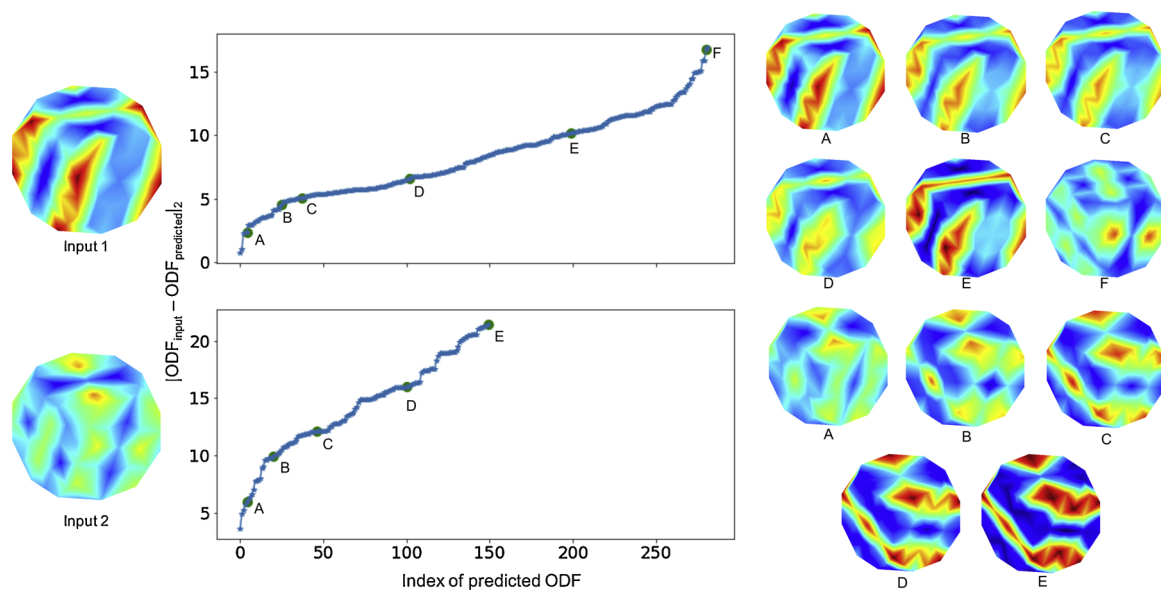

Fig. 8. Two cases to illustrate the sequence prediction task through ODF identification. ODFs on the right are a few matched points from the training set, which have varying L2 errors with respect to the input, indicated by numbers on the error curve.

Table 3
Process sequences for the input cases from Fig. 8.

	Stage 1	β_1	Stage 2	β_2	Stage 3	β_3	Stage 4	β_4
1	Tension Z	1.0	Compression X	0.5	Compression Z	1.0	Rolling XZ	-0.5
2	Tension X	1.0	Rolling ZY	-0.5	Rolling XZ	0.5	Compression Y	1.0

performance by looking for the preservation of important fibers from Rodrigues space. For this purpose three ODFs are used as inputs to different models, and lines shown on these are the important fibers whose preservation is an important metric of model performance. From ODF 1, it is seen that all the models are able to capture the most important features, but with varying intensity matching. From ODF 2, dim = 2 captures the top layer marked as a single fibre whereas the

other models are able to replicate the input ODF better. In ODF 3, looking at the red lines, it is again not sufficiently reproduced in dim = 2 and it gets better with increasing dimensions. Though dim = 1 is able to capture many features, it does not sufficiently show diversity in the patterns.

Together, Fig. 5 shows that the most optimal choice of latent space dimension is 3, considering the trade-off between size and performance.

Table 4
Process sequences predicted for the first test case from Fig. 8.

	Stage 1	β_1	Stage 2	β_2	Stage 3	β_3	Stage 4	β_4
A	Tension Y	0.5	Tension Z	1.0	Tension X	0.5	Rolling XZ	1.0
B	Rolling XZ	1.0	Tension Y	−1.0	Tension Y	1.0	Tension Z	0.5
C	Rolling ZY	0.5	Rolling XZ	1.0	Tension Z	1.0	NaN	NaN
D	Tension Y	0.5	Rolling XZ	1.0	Rolling ZY	−0.5	NaN	NaN
E	Rolling XZ	0.5	Tension X	−1.0	Tension Y	−0.5	NaN	NaN
F	Tension X	−1.0	Tension Y	0.5	Rolling ZY	−1.0	Rolling XZ	−0.5

Table 5
Process sequences predicted for the second test case from Fig. 8.

	Stage 1	β_1	Stage 2	β_2	Stage 3	β_3	Stage 4	β_4
A	Compression X	0.5	Rolling ZY	−1.0	Tension X	1.0	Compression Z	0.5
B	Rolling YX	1.0	Rolling ZY	−0.5	Rolling XZ	−0.5	NaN	NaN
C	Tension Z	1.0	Compression Z	1.0	Tension X	1.0	Compression Y	1.0
D	Tension X	0.5	Compression Y	1.0	Rolling YX	0.5	Tension X	0.5
E	Rolling XZ	−1.0	Compression Y	1.0	Rolling YX	1.0	Rolling YX	1.0

A 2D latent space performs nearly close to a 3D latent space in most of the cases. Since this paper aims at understanding the database through a graphical treatment, and the spatial separation in the latent space, the results going further are shown from a 2D latent space, with one example of prediction from a 3D latent space.

The approach to finding a process sequence for an unknown ODF involves mapping this ODF onto the latent space and identifying other ODFs in the vicinity. In that vein, it would be necessary to understand the similarity of ODFs in a small window in the latent space, and the processes which resulted in those particular structures. Fig. 6 shows the ODFs in one such circular window with radius 0.1 units. It is seen that all ODFs identified in the area, exhibit very similar features and fall into two different sets of similar ODFs. Table 2 shows the processes that gave rise to these textures and the effective elastic modulus in x direction (E_x). The properties are obtained as a polycrystal average by an integral over the fundamental region: $\bar{C} = \int_{\mathcal{R}} C \, \mathcal{A}(\mathbf{r}) \, \mathrm{dv}$. The stiffness constants for FCC Aluminum are used here. The elastic modulus is computed as $E = \frac{1.0}{(\bar{C})_{(11)}^{-1}}$. Even though the ODFs exhibit very similar features, their E_x values show a significant spread. This can be attributed to small variations at certain nodes which arise from distinct processes and their order as well. Thus, using this strategy, it is possible to identify ODFs and sequences which are similar, but also identify process sequences and resultant textures which give rise to better properties.

Fig. 7a shows ten ODFs from the test set mapped onto the latent space. The region shows latent coordinates of ODFs from the training set. Next to each ODF, there is a cluster of points which represent ODFs in close proximity to the input ODF, as described previously. These points are colored by the difference between E_x of input ODF and ODFs to which they correspond. For simplicity these ODFs will be called 'predicted ODFs' going forward. Fig. 7b shows L2 error between each input ODF and its corresponding predicted ODFs. The number of predicted ODFs depends on the density of points in the latent space, and since the density closer to origin is higher, test cases 1,4, and 8 have the highest number of predicted ODFs.

To illustrate sequence identification, the predicted ODFs for inputs 5 and 7 from Fig. 7 are examined. These will be referred to as inputs 1 and 2 going forward. Fig. 8 shows the two inputs, along with 6 and 5 predicted ODFs for inputs 1 and 2 respectively. The L2 difference between input and each predicted ODF are highlighted by green dots in both cases. Even though a few ODFs are exactly similar to the input, the L2 difference is not zero because of small perturbations. Tables 3–5 show the process paths that led to these ODFs. The tables show that the

model is capable of identifying ODFs which have resulted from sequences with lesser stages for the illustrated cases. While 1A, 1B, and 1C are all identically similar to the input, 1D, 1E, and 1F are not. The latter three have been identified because they preserve the important features of input, though they have different intensities. 1B, 1C, and 1D have only 3 stages in their respective processes, while the input has 4 stages.

The database and trained model can also be used for material design by finding new ODFs in the neighborhood with approximately same property of interest. Fig. 9 shows the same two inputs from the previous case, but the plot shows difference between E_x of input ODFs and their respective predicted ODFs. Also shown are two predicted ODFs for each case whose E_x is very close to their respective inputs, and these points correspond to the blue and red dots on respective curves. It is interesting to see that for both the cases, the ODFs shown are not similar to the inputs. There is either a difference in the features or the intensity. But, the difference in E_x is less than 0.5 GPa. The graphs also show that there can be multiple ODFs from the $|\Delta E_x| < 1$ GPa range to choose for predicted ODFs.

For visualizing the distribution of ODFs and for illustrating proximity, using a 2D latent space is more useful. But, from the latent
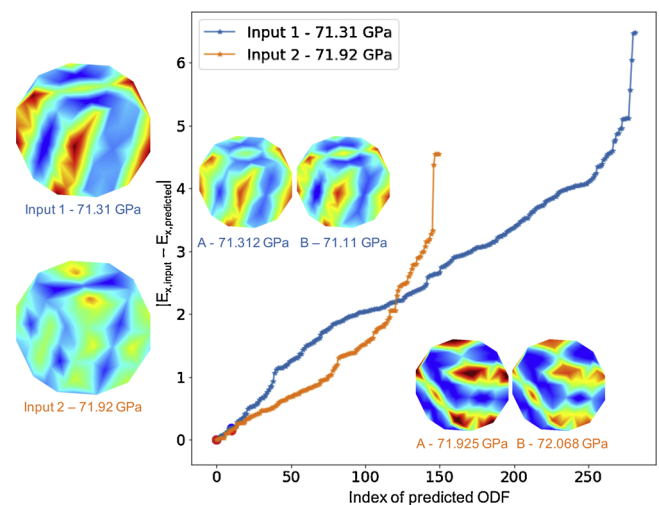
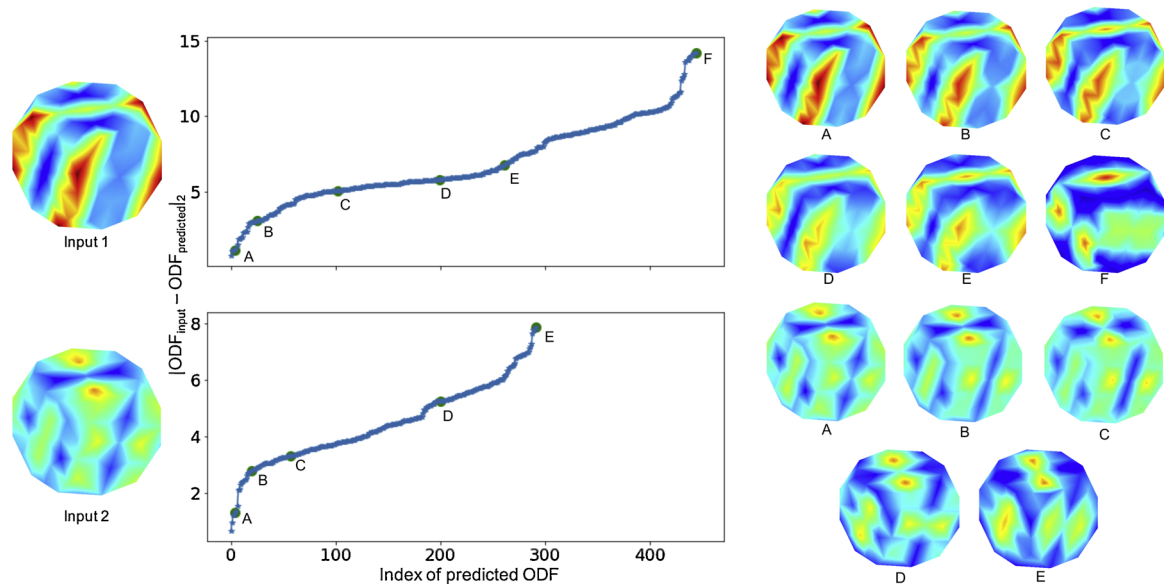**Fig. 9.** ODF identification by matching E_x between input and prediction.

Fig. 10. ODF identification task from a 3D latent space.

dimension comparison exercise, a 3D latent space was identified to be most optimal for reconstruction and feature capture. Thus, the prediction task was carried out using a model with a 3D latent space as shown in Fig. 10. Similar to the circular area used for the previous case, here the ODFs within a spherical volume with radius 0.2 units from the input ODF were selected for prediction. While the cut-off is arbitrarily chosen, number of predicted ODFs for both inputs is higher. Despite this, the maximum L2 error is lesser compared to a 2D latent space for both inputs.

4. Conclusions

The database conceived and disseminated here is a first step in the efforts to unearth PSP linkages. A few features and uses of this database has been explored here. A visual treatment of the database was presented using a VAE for dimensionality reduction through a latent space. The latent space representation exhibits a spatial distribution of ODF based on important features. Moreover, nearly identical ODFs are mapped to the same locations in this space. By searching in the vicinity of an ODF mapped in the latent space for other similar ODFs, it has been shown that it is possible to identify processing routes. A few of these processing routes were also shown to produce similar ODFs in using fewer process stages. Further work will involve training the VAE to use process sequences as an input to the model. This will then be used to identify new process sequences for ODFs identified from the latent space through interpolatory and extrapolatory modes. In the field of natural language processing, sequence to sequence prediction (seq2seq) is a technique used to predict a new word in a sentence, given the previous words [30]. This technique can be used to iteratively predict each stage in a sequence and this will also be explored in the future.

Acknowledgements

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.mtcomm.2020.101201.

References

[1] A. Paul, P. Acar, R. Liu, W. Liao, A. Choudhary, V. Sundararaghavan, A. Agrawal, Data sampling schemes for microstructure design with vibrational tuning constraints, AIAA J. 56 (3) (2018) 1239–1250.

[2] A. Paul, P. Acar, W.-k. Liao, A. Choudhary, V. Sundararaghavan, A. Agrawal, Microstructure optimization with constrained design objectives using machine learning-based feedback-aware data-generation, Comput. Mater. Sci. 160 (2019) 334–351.

[3] A. Cecen, H. Dai, Y.C. Yabansu, S.R. Kalidindi, L. Song, Material structure–property linkages using three-dimensional convolutional neural networks, Acta Mater. 146 (2018) 76–84.

[4] J. Jung, J.I. Yoon, H.K. Park, J.Y. Kim, H.S. Kim, An efficient machine learning approach to establish structure–property linkages, Comput. Mater. Sci. 156 (2019) 17–25.

[5] B.L. DeCost, T. Francis, E.A. Holm, Exploring the microstructure manifold: image texture representations applied to ultrahigh carbon steel microstructures, Acta Mater. 133 (2017) 30–40.

[6] D.T. Fullwood, S.R. Niezgoda, B.L. Adams, S.R. Kalidindi, Microstructure sensitive design for performance optimization, Prog. Mater. Sci. 55 (6) (2010) 477–562.

[7] H. Xu, D.A. Dikin, C. Burkhart, W. Chen, Descriptor-based methodology for statistical characterization and 3d reconstruction of microstructural materials, Comput. Mater. Sci. 85 (2014) 206–216.

[8] M. Mozaffar, R. Bostanabad, W. Chen, K. Ehmann, J. Cao, M. Bessa, Deep learning predicts path-dependent plasticity, Proc. Natl. Acad. Sci. 116 (52) (2019) 26414–26420.

[9] R. Liu, A. Kumar, Z. Chen, A. Agrawal, V. Sundararaghavan, A. Choudhary, A predictive machine learning approach for microstructure optimization and materials design, Sci. Rep. 5 (11551) (2015) 1–12.

[10] E. Popova, T.M. Rodgers, X. Gong, A. Cecen, J.D. Madison, S.R. Kalidindi, Process-structure linkages using a data science approach: application to simulated additive manufacturing data, Integr. Mater. Manuf. Innov. 6 (1) (2017) 54–68, https://doi.org/10.1007/s40192-017-0088-1.

[11] J. Jung, J.I. Yoon, S.-J. Park, J.-Y. Kang, G.L. Kim, Y.H. Song, S.T. Park, K.W. Oh, H.S. Kim, Modelling feasibility constraints for materials design: application to inverse crystallographic texture problem, Comput. Mater. Sci. 156 (2019) 361–367.

[12] P. Acar, V. Sundararaghavan, Linear solution scheme for microstructure design with process constraints, AIAA J. 54 (12) (2016) 4022–4031.

[13] P. Acar, V. Sundararaghavan, Reduced-order modeling approach for materials design with a sequence of processes, AIAA J. 56 (12) (2018) 5041–5044.

[14] D. Li, H. Garmestani, B. Adams, A texture evolution model in cubic-orthotropic polycrystalline system, Int. J. Plasticity 21 (8) (2005) 1591–1617.

[15] D. Li, H. Garmestani, S. Ahzi, Processing path optimization to achieve desired texture in polycrystalline materials, Acta Mater. 55 (2) (2007) 647–654.

[16] D.B. Brough, D. Wheeler, J.A. Warren, S.R. Kalidindi, Microstructure-based knowledge systems for capturing process-structure evolution linkages, Curr. Opin. Solid State Mater. Sci. 21 (3) (2017) 129–140.

[17] J.B. Shaffer, M. Knezevic, S.R. Kalidindi, Building texture evolution networks for deformation processing of polycrystalline fcc metals using spectral approaches: applications to process design for targeted performance, Int. J. Plasticity 26 (8) (2010) 1183–1194.

[18] A. Khosravani, A. Cecen, S.R. Kalidindi, Development of high throughput assays for establishing process-structure-property linkages in multiphase polycrystalline metals: application to dual-phase steels, Acta Mater. 123 (2017) 55–69.

[19] Y.C. Yabansu, P. Steinmetz, J. Hötzer, S.R. Kalidindi, B. Nestler, Extraction of reduced-order process-structure linkages from phase-field simulations, Acta Mater. 124 (2017) 182–194.

[20] V. Sundararaghavan, N. Zabaras, On the synergy between texture classification and deformation process sequence selection for the control of texture-dependent properties, Acta Mater. 53 (4) (2005) 1015–1027.

[21] V. Sundararaghavan, N. Zabaras, A statistical learning approach for the design of polycrystalline materials, Stat. Anal. Data Min. 1 (5) (2009) 306–321.

[22] A. Kumar, P. Dawson, Computational modeling of fcc deformation textures over Rodrigues' space, Acta Mater. 48 (10) (2000) 2719–2736.

[23] M.A. Kramer, Nonlinear principal component analysis using autoassociative neural networks, AIChE J. 37 (2) (1991) 233–243, https://doi.org/10.1002/aic.690370209.

[24] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016, http://www.deeplearningbook.org.

[25] D.P. Kingma, M. Welling, Auto-Encoding Variational Bayes, (2013) arXiv:1312.6114.

[26] C. Doersch, Tutorial on Variational Autoencoders, (2016) arXiv:1606.05908.

[27] R. Gómez-Bombarelli, J.N. Wei, D. Duvenaud, J. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T.D. Hirzel, R.P. Adams, A. Aspuru-Guzik, Automatic chemical design using a data-driven continuous representation of molecules, ACS Cent. Sci. 4 (2) (2018) 268–276, https://doi.org/10.1021/acscentsci.7b00572.

[28] H.S. Stein, D. Guevarra, P.F. Newhouse, E. Soedarmadji, J.M. Gregoire, Predicting spectra from images and images from spectra, Chem. Sci. 10 (1) (2018), https://doi.org/10.22002/D1.1103.

[29] J. Noh, J. Kim, H.S. Stein, B. Sanchez-Lengeling, J.M. Gregoire, A. Aspuru-Guzik, Y. Jung, Inverse design of solid-state materials via a continuous representation, Matter 1 (5) (2019) 1370–1384, https://doi.org/10.1016/j.matt.2019.08.017.

[30] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, Advances in Neural Information Processing Systems (2014) 3104–3112.