A multi-task learning-based optimization approach for finding diverse sets of microstructures with desired properties

Tarek Iraki[1] · Lukas Morand[2] · Johannes Dornheim[3] · Norbert Link[1] · Dirk Helm[2]

Abstract

Optimization along the chain processing-structure-properties-performance is one of the core objectives in data-driven materials science. In this sense, processes are supposed to manufacture workpieces with targeted material microstructures. These microstructures are defined by the material properties of interest and identifying them is a question of materials design. In the present paper, we address this issue and introduce a generic multi-task learning-based optimization approach. The approach enables the identification of sets of highly diverse microstructures for given desired properties and corresponding tolerances. Basically, the approach consists of an optimization algorithm that interacts with a machine learning model that combines multi-task learning with siamese neural networks. The resulting model (1) relates microstructures and properties, (2) estimates the likelihood of a microstructure of being producible, and (3) performs a distance preserving microstructure feature extraction in order to generate a lower dimensional latent feature space to enable efficient optimization. The proposed approach is applied on a crystallographic texture optimization problem for rolled steel sheets given desired properties.

Introduction

Motivation

The demand for more and more specific and individually designed products with certain performance requirements has become a driving force in the world of manufacturing. For this reason, the optimization along the causal chain processing-structure-properties-performance (Olson, 1997) became a fast growing research topic in the field of integrated computational materials engineering (ICME) (Panchal et al., 2013). Nowadays, such optimization problems can be solved efficiently with the help of machine learning techniques (Ramprasad et al., 2017). On this background, in a previous work, we investigated the use of reinforcement learning for finding optimal processing routes in a simulated metal forming process aiming to produce microstructures with targeted crystallographic textures (Dornheim et al., 2021). To bridge the remaining gap between microstructures and desired properties, we focus in this work on solving materials design problems. These are to identify appropriate material microstructures or microstructural features (e.g. the crystal-

✉ Tarek Iraki
  tarek.iraki@h-ka.de

  Lukas Morand
  lukas.morand@iwm.fraunhofer.de

  Johannes Dornheim
  johannes.dornheim@kit.edu.de

  Norbert Link
  norbert.link@h-ka.de

  Dirk Helm
  dirk.helm@iwm.fraunhofer.de

1   Intelligent Systems Research Group (ISRG), Karlsruhe University of Applied Sciences, Moltkestr. 30, 76133 Karlsruhe, Germany

2   Fraunhofer Institute for Mechanics of Materials IWM, Wöhlerstraße 11, 79108 Freiburg, Germany

3   Institute for Applied Mechanics - Computational Materials Sciences IAM-CMS, Karlsruhe Institute of Technology, Karlsruhe, Germany

lographic texture) for given desired properties. It is thereby of particular importance to identify sets of near-optimal and preferably diverse microstructures in order to guarantee a robust design (McDowell, 2007).

Paper structure

In the following we summarize the related work and point out the contribution of this paper to the actual research. In "Methods" section, first, we describe the siamese multi-task learning and optimization approach. Then, we introduce the fundamentals in materials modeling that are needed for the purpose of this work. After that, in "Results" section, the results are shown when applying the approach to a texture optimization problem for steel sheets (in particular, we fit the material model to DC04 steel). In "Discussion" section, the presented results are discussed. Finally, in "Summary and Outlook" section, we summarize our findings and give an outlook on further research.

Related work

A generic approach to solve materials design problems is the microstructure sensitive design (MSD) approach introduced in Adams et al. (2001). Following Fullwood et al. (2010), MSD can be described by seven steps. First, the properties of interest as well as candidate materials have to be defined. After that, a suitable microstructure definition is applied for these materials yielding a microstructure design space. On this basis, relevant homogenization relations are identified and applied over the whole design space. The resulting property closure can be used to select desired properties, which are then mapped back to the microstructure design space in order to identify optimal microstructures. The last step of MSD aims to determine processes and processing routes needed to produce the identified microstructure.

The works by Adams et al. (2001) and Kalidindi et al. (2004) instantiate the MSD approach for texture optimization. The first one describes how optimal crystallographic textures can be identified in order to improve the deformation behavior of a compliant beam. In the latter, a similar approach is shown to optimize the crystallographic texture for the design of an orthotropic plate. The core of both approaches lies in the usage of a lower dimensional spectral representation of the orientation distribution, cf. Bunge (2013). For more complex microstructure representations, like two-point correlations, feature extraction methods can be applied to reduce the dimensionality. Methods that are used in the context of materials design are principal component analysis (PCA) (Paulson et al., 2017; Gupta et al., 2015) and multidimensional scaling (Jung et al., 2019) for example. A general review of dimensionality reduction techniques can be found in Van Der Maaten et al. (2009).

Besides the MSD approach, also machine learning-based approaches for crystallographic texture optimization exist. Liu et al. (2015) and Paul et al. (2019) describe iterative sampling approaches that interact with crystal plasticity simulations aiming to identify crystallographic textures for given desired properties. Therefore, an initial set of texture-property tuples (crystallographic textures and corresponding macroscopic properties) is generated. Via supervised learning, significant features of the parameterized orientation distribution (and in Liu et al. (2015) also significant regions) are identified that yield optimal or near-optimal solutions. Based on the identified features and regions, new texture-property data points are sampled in order to get closer to the optima.

Another approach for identifying optimal textures is described in Kuroda and Ikawa (2004). Therein, a real-coded genetic algorithm (Goldberg, 1991) is described that interacts with a crystal plasticity model in order to find optimal combinations of typical rolling texture components of face-centered cubic metal (Cu, Brass, S, Cube and Goss) for given desired properties. The algorithm starts with an initial set of textures consisting of different fractions of these components. The set of textures evolves iteratively by combining them using operators such as mutation, crossover and selection (Herrera et al., 1998).

Recent works (i.e., Jung et al. (2020) and Kamijyo et al. (2022)) use Bayesian optimization for microstructure design. In Kamijyo et al. (2022), a deep neural network is used for the estimation of mechanical properties. On this basis, Bayesian optimization is used to determine optimal volume fractions of texture components of aluminum (cf. Kuroda and Ikawa (2004)) for a desired formability. For designing complex microstructure models, in Jung et al. (2020), the use of the latent space of a convolutional autoencoder as a low dimensional design space is proposed. Within this design space, Bayesian optimization is adopted to search for optimal dual phase microstructures for given desired properties (i.e, tensile strength).

Predicting dual phase microstructure properties using convolutional neural networks (CNN) is also used in Mann and Kalidindi (2022), however, to explore the properties space defined by the material stiffness. The CNN architecture was developed for approximating the highly non-linear microstructure–property linkage, while using also two-point spatial correlation functions of the microstructure as input.

A further convolutional approach is described in Tan et al. (2020), in which a deep convolutional generative adversarial network (DCGAN) and a CNN is proposed for the design of materials. Therein, the CNN links the micostructures to its properties and acts as a surrogate model, whereas the DCGAN generates design candidates for a desired compliance tensor.

Summarized, for the solution of microstructure design problems, a linkage from properties to microstructures is required. Such a linkage is often achieved by genetic or optimization algorithms that interact with numerical simulations. However, as these algorithms generally need a lot of function evaluations, it is not reasonable to apply them to complex numerical simulations directly. Instead, the performance can be increased by using numerically simpler surrogate models, see for example (Simpson et al., 2001). Typically, these are supervised learning models that learn the input–output relations of the numerical simulation under consideration.

To run optimization algorithms in combination with supervised learning models it is necessary to limit the region in which they operate to the region, which is covered by the training data. One way to achieve this is by training unsupervised learning models on the input data, as it is done in Jung et al. (2019) for example using support vector machines (SVM). From a machine learning perspective such an approach can be seen as anomaly detection. Anomaly detection aims to separate data that is characteristically different from the known data of the sample data set, which has been used for training. An extensive overview of anomaly detection methods is given in Chandola et al. (2009). Moreover, Ruff et al. (2021) and Chalapathy and Chawla (2019) gives an overview on recent deep learning-based approaches for anomaly detection, from which we want to point out neural network-based autoencoders (Hinton & Salakhutdinov, 2006), which fit especially well into multi-task learning (MTL) (Caruana, 1997) schemes other than SVMs.

Autoencoder approaches assume that features of a data set can be mapped into a lower dimensional latent feature space, in which the known data points differ substantially from unknown data points. By backmapping into the original space, anomalies can be identified by evaluating the reconstruction error, see for example Sakurada and Yairi (2014). In Sakurada and Yairi (2014) it is also shown that autoencoder networks are able to detect subtle anomalies, which cannot be detected by linear methods like PCA. Furthermore, autoencoder networks require less complex computations compared to a nonlinear kernel-based PCA.

Recent developments in anomalie detection include deep learning approaches, like the deep support vector data description method (Deep SVDD) (Ruff et al., 2018). Deep SVDD is an unsupervised anomaly detection method, which is inspired by kernel-based one-class classification and minimum volume estimation, and can be traced back to traditional methods, which are One-Class SVM (Schölkopf et al., 2001) and SVDD (Tax and Duin, 2004). In contrast to autoencoder approaches, Deep SVDD is based on an anomaly detection objective, rather than relying on the reconstruction error. By using Deep SVDD, a neural network is trained while minimizing the volume of a hypersphere that encloses the network representations of the data. By minimizing the objective,

Deep SVDD aims to find a preferably small data-enclosing hypersphere and learns to extract the common factors of variation of the data distribution. The aim of the approach is that representations of the normal data lie inside the hypersphere, while anomalous data points lie outside the hypersphere. Thereby, anomalies can be detected based on their distance to the centroid of the hypersphere.

An extension of Deep SVDD is given by the method Improved AutoEncoder for Anomaly Detection (IAEAD) (Cheng et al., 2021) by combining Deep SVDD with autoencoders. The autoencoder is used for the embedding of the features and to preserve the local structure of the data generating distribution, whereas Deep SVDD detects anomalies in the feature space. This is achieved by adding the Deep SVDD loss as a regularization term to the original autoencoder optimization objective (i.e. the minimization of the reconstruction error). However, instead of using the reconstruction error, IAEAD uses the distance to the centroid in feature space for anomaly detection like the original Deep SVDD approach.

Another recently developed approach uses an autoencoder at the example of learning image data by minimizing the reconstruction error (defined as the loss function) (Kwon et al., 2020). The trained model is used for anomaly detection by evaluating the gradients of the reconstruction error with respect to the neural network weights. Gradients are generated through backpropagation to train neural networks by minimizing designed loss functions (Rumelhart et al., 1986). While feeding new input data into to the neural network, the gradients originating from normal data cause only slight changes with respect to the neural network weights, whereas the gradients from anomalous data cause more drastic changes. Thus, anomalies can be detected by measuring how much of the input data does not correspond to the learned information of the network in terms of the gradients.

Summary of related work and contribution

Optimizing the crystallographic texture of sheet metal has been studied in various publications. So far, classic optimization approaches exist, that operate on well-established crystallographic texture representations from the field of materials science (i.e. using texture components or a spectral decomposition). In addition, machine learning-based approaches have been developed in order to efficiently guide optimization algorithms to promising regions in texture space.

Regarding microstructure optimization in general, the usage of machine learning models has become popular during the last years. Often supervised learning models are used to learn and replace time-consuming numerical simulations for propertiy prediction. Furthermore, unsupervised learning models (often PCA) are used reduce the dimensionality

of complex microstructure representations. In the field of machine learning, however, more sophisticated approaches exist, such as nonlinear methods for feature extraction and MTL approaches that combine different learning tasks into one model with the advantage of having a universal latent feature space.

For the processing of optimal microstructures (which is the next step in the processing-structure-properties chain), it is advantageous to identify not only one optimal microstructure for given desired properties, but a set of nearly optimal microstructures that is as diverse as possible. Such approaches are, however, lacking in literature. A further important requirement for optimal processing not much addressed in microstructure optimization approaches is to consider the producibility of identified microstructures.

Therefore, in the present paper, we introduce a generic MTL-based optimization approach to efficiently identify sets of microstructures, which are highly diverse and guaranteed to be producible by a dedicated manufacturing process. The approach is based on an optimization algorithm interacting with a machine learning model that combines MTL with siamese neural networks (Bromley et al., 1993). In contrast to Liu et al. (2015), Paul et al. (2019) and also to Kuroda and Ikawa (2004), in our approach, a surrogate model is set up in order to replace the numerical simulation, which maps microstructures to properties. The microstructure–property mapping can be executed efficiently by means of the surrogate model within the optimization procedure.

In order to increase the efficiency of the optimization approach, the microstructure representation is transformed into a lower dimensional latent feature space by a non-linear data-driven encoder. The encoder in turn provides the input signal for three attached learning tasks of the MTL-approach. The first learning task maps the features to properties (surrogate model). To address the issue of producibility, we include a second learning task, which estimates the validity of a microstructure in the sense of being producible (being part of the region enclosed by the underlying data set). The third learning task is the decoder for the microstructure representation.

As learning takes place simultaneously for the encoder and the attached tasks, it is ensured that the lower dimensional feature space is optimal for all tasks. In addition, we enforce the latent feature space to preserve microstructure distances by employing a siamese neural network and multidimensional scaling. On this basis, we force the optimizer to find a diverse set of optimal microstructures in the latent feature space.
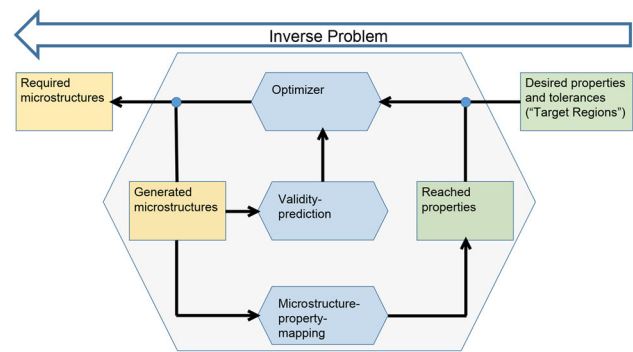

Fig. 1 General concept of the MTL-based optimization approach to solve materials design problems

Methods

Materials design via siamese multi-task learning (SMTL) and optimization

General concept

First of all, we present the general concept of our MTL-based optimization approach. The approach can be applied to general materials design problems and starts by defining the desired properties and corresponding tolerances. This in turn defines a target region, for which the approach is supposed to identify a diverse set of microstructures. The approach is schematically depicted in Fig. 1 and summarized in Table 1. It basically consists of three components: optimizer, microstructure–property mapping (*m-p-m*) and validity-prediction (*v-p*). The optimizer generates candidate microstructures that minimize the combined costs, which result from evaluations based on the *m-p-m* and *v-p* components.

The *m-p-m* component assigns properties to a candidate microstructure. The deviation of the assigned properties to the target region determines the cost. In general, the *m-p-m* component can be realized by a numerical simulation. However, since numerical simulations are computationally expensive, a surrogate model is used instead. The surrogate model is realized by a regression model that learns the relations from a priori generated microstructure–property data.

The *v-p* component is realized by an anomaly detection method which determines the validity of a candidate microstructure by comparing it to the set of valid microstructures. The concept of the anomaly detection is illustrated in Fig. 2. The *v-p* component returns a value that can be seen as an estimate of a candidate microstructure being an element of the microstructure set under consideration. This is for example the set, which can be produced by a dedicated process (e.g. rolling). The value returned by the *v-p* component defines the validity cost and is supposed to drive the

Table 1 Summary of important symbols used throughout this study

| Notation	Description
Variables	
$\boldsymbol{x} \in \mathbb{R}^K$	Microstructure representation
$\boldsymbol{x}' \in \mathbb{R}^K$	Reconstructed microstructure
$\boldsymbol{p} \in \mathbb{R}^N$	True properties from the dataset
$\hat{\boldsymbol{p}} \in \mathbb{R}^N$	Predicted properties by the reggression network
$\boldsymbol{z} \in \mathbb{R}^M$	Low dimensional latent feature
$\boldsymbol{z}' \in \mathbb{R}^M$	Reconstructed low dimensional latent feature
$\boldsymbol{s} \in \mathbb{R}^S$	Low dimensional latent feature in validity network
f(g)	Orientation distribution function
$\boldsymbol{b}$	Orientation histogram
$\chi^2(\boldsymbol{b}_1, \boldsymbol{b}_2)$	Chi-Squared distance between two orientation histograms
$\boldsymbol{T}^{(i)}$	Cauchy stress of ith crystal
$\boldsymbol{F}$	Deformation gradient
$\boldsymbol{L}$	Velocity gradient
E_{00}, E_{45}, E_{90}	Young's modulus in 0, 45 and 90 degree to rolling direction
r_{00}, r_{45}, r_{90}	R-value in 0, 45 and 90 degree to rolling direction
Functions	
m-p-m	Microstructure–property mapping
v-p	Validity-prediction
$f_{\text{enc}}(\boldsymbol{x}, \boldsymbol{\theta}_{\text{enc}})$	Encoder network
$f_{\text{regr}}(\boldsymbol{z}, \boldsymbol{\theta}_{\text{regr}})$	Regression network
$f_{\text{recon}}(\boldsymbol{z}, \boldsymbol{\theta}_{\text{recon}})$	Reconstruction network
$f_{\text{valid}}(\boldsymbol{z}, \boldsymbol{\theta}_{\text{valid}})$	Validity network
$\boldsymbol{\theta}_{\text{enc}}$	Weight values for the encoder network
$\boldsymbol{\theta}_{\text{regr}}$	Weight values for the regression network
$\boldsymbol{\theta}_{\text{recon}}$	Weight values for the reconstruction network
$\boldsymbol{\theta}_{\text{valid}}$	Weight values for the validity network
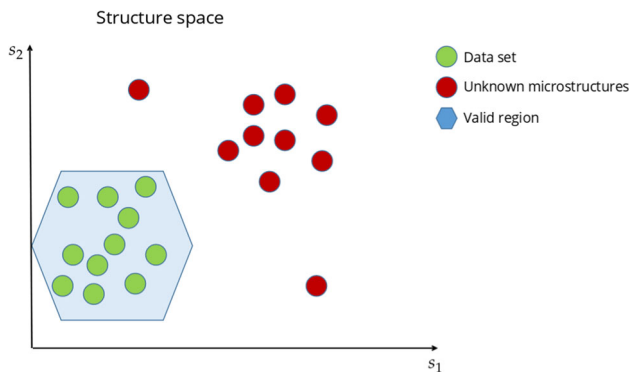
Fig. 2 Schematic illustration of the structure space s_1, s_2 including the valid region and unknown microstructures

optimizer solution to a valid microstructure region. Besides, such a microstructure region can alternatively be identified by a further optimization loop that interacts with a numerical simulation of the dedicated process, which, however, suffers from high computational costs.

The two components *m-p-m* and *v-p* can be realized by training two separate machine learning models. However, when the training procedures are isolated from each other, the models are not able to mutually access information already learned by the other model. Therefore, we combine the two components as tasks into one MTL model (Caruana, 1997). Both tasks have a common backbone (the feature extraction part of a network) and different heads (feature processing part of a network) operating on the backbone output. The backbone output vectors form the so-called latent feature space.

The proposed MTL approach furthermore uses the backbone as an encoder network of an autoencoder, where the decoder is also attached to the latent feature space with the purpose to reconstruct the input pattern of the backbone. This is achieved by adding the reconstruction of the microstructures from the latent feature space as a third task. In the MTL approach, all the three tasks are represented by a single neural network-based model. The neural weights of the model are trained simultaneously based on a combined loss function.

After training the MTL model, the optimizer can operate very efficiently in the lower dimensional latent feature space.

However, it has some limitations, which are mentioned in the following. Since our concept is based on a data-driven modeling (machine learning) and optimization approach, an adequate data set is required. The described components which are learned within the concept are approximations of the numerical simulations and are accordingly not equally exact. The model quality of the components depends on the size and quality of the underlying data set. Therefore, the application of an efficient sampling strategy for exploring the microstructure and property space can be suitable (Morand et al., 2022). However, under the assumption of low model errors, the components can be efficiently used as surrogate models in the application of the concept (except for extrapolation).

The remainder of this section presents the optimization approach and the MTL approach in detail, as well as an extension based on siamese neural networks (Bromley et al., 1993) to enforce the representation of microstructures in the latent feature space to preserve the microstructure distances in the original representation space.

Multi-task learning (MTL)

The MTL model, as shown in Fig. 3, is trained on pairs of microstructures and corresponding properties $(\boldsymbol{x}, \boldsymbol{p})$. The input microstructures are transformed into latent features $\boldsymbol{z}$. The individual outputs of the connected tasks are the estimated properties $\hat{\boldsymbol{p}}$, the reconstructed microstructure $\boldsymbol{x}'$ and the reconstructed latent features $\boldsymbol{z}'$. In the following, we introduce the information processing scheme of the MTL model in detail.

The processing scheme starts with an encoder network which extracts significant features by mapping the microstructure space $\boldsymbol{x} \in \mathbb{R}^K$ into a lower dimensional latent feature space $\boldsymbol{z} \in \mathbb{R}^M$ via the learned function

$$z = f_{\text{enc}}(\boldsymbol{x}, \boldsymbol{\theta}_{\text{enc}}), \tag{1}$$

in which the encoder network is parameterized by its weight values $\boldsymbol{\theta}_{\text{enc}}$. All three previously described tasks are attached to the encoder in the form of feedforward neural networks. Besides, the encoder can be easily adapted to higher dimensional microstructure representing data types, like images (EBSD or micrograph images) or three dimensional microstructure data by using convolutional neural networks (see Krizhevsky et al. (2012)). The latter is used in Cecen et al. (2018) in the materials sciences domain for example.

To train the MTL model, a loss function that combines all the three tasks is needed. This is achieved by a function that cumulates the loss terms of the three tasks $\mathcal{L}_{\text{regr}}$ (regression loss), $\mathcal{L}_{\text{recon}}$ (reconstruction loss) and $\mathcal{L}_{\text{valid}}$ (validity loss),
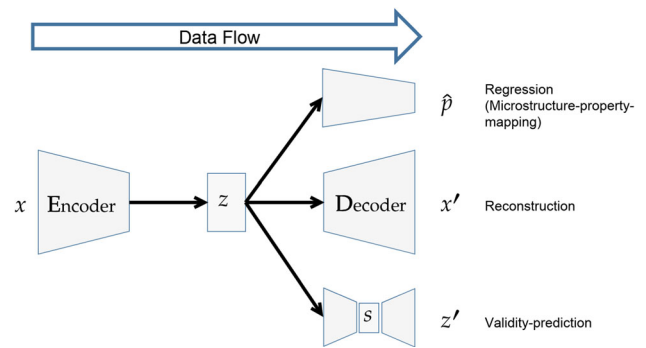

Fig. 3 MTL architecture

and weights them using $\mathcal{W}_{\text{regr}}$, $\mathcal{W}_{\text{recon}}$ and $\mathcal{W}_{\text{valid}}$ to allow for prioritization. The total loss function is defined as

$$
\begin{aligned}
\mathcal{L}_{\text{MTL}} = {} & \mathcal{W}_{\text{regr}}\mathcal{L}_{\text{regr}} + \mathcal{W}_{\text{recon}}\mathcal{L}_{\text{recon}} \\
& + \mathcal{W}_{\text{valid}}\mathcal{L}_{\text{valid}} + \lambda R(\boldsymbol{\theta}),
\end{aligned}
\tag{2}
$$

where $R(\boldsymbol{\theta})$ is a regularization term that is used to prevent overfitting with the hyperparameter λ defining the strength of the regularization (also known as weight decay, see Krogh and Hertz (1991) and Hinton (1987)). Each of the feedforward neural networks is parameterized by the respective weight values $\boldsymbol{\theta}_{\text{enc}}$, $\boldsymbol{\theta}_{\text{regr}}$, $\boldsymbol{\theta}_{\text{recon}}$ and $\boldsymbol{\theta}_{\text{valid}}$, which are adjusted simultaneously during training and altogether form the weight vector $\boldsymbol{\theta}$. In the following we introduce the three individual loss terms.

1. The forward mapping of the latent feature vector $\boldsymbol{z}$ to the properties vector $\hat{\boldsymbol{p}} \in \mathbb{R}^N$ is represented by the learned function

$$\hat{\boldsymbol{p}} = f_{\text{regr}}(\boldsymbol{z}, \boldsymbol{\theta}_{\text{regr}}) = f_{\text{regr}}(f_{\text{enc}}(\boldsymbol{x}, \boldsymbol{\theta}_{\text{enc}}), \boldsymbol{\theta}_{\text{regr}}). \tag{3}$$

The regression loss is given by the mean squared error between the predicted properties $\hat{\boldsymbol{p}}$ and the true properties $\boldsymbol{p}$:

$$\mathcal{L}_{\text{regr}}(\boldsymbol{p}, \hat{\boldsymbol{p}}) = \frac{1}{N}\sum_{i=1}^{N}(p_i - \hat{p}_i)^2, \tag{4}$$

where N denotes the number of properties.

2. The decoder network, which is responsible for the reconstruction, transforms the latent feature vectors $\boldsymbol{z}$ back to the original microstructure space:

$$\boldsymbol{x}' = f_{\text{recon}}(\boldsymbol{z}, \boldsymbol{\theta}_{\text{recon}}) = f_{\text{recon}}(f_{\text{enc}}(\boldsymbol{x}, \boldsymbol{\theta}_{\text{enc}}), \boldsymbol{\theta}_{\text{recon}}). \tag{5}$$

The reconstruction loss is defined on the basis of a distance measure between two microstructural feature

vectors $\text{dist}(\boldsymbol{x}, \boldsymbol{x}')$:

$$\mathscr{L}_{\text{recon}}(\boldsymbol{x}, \boldsymbol{x}') = \text{dist}(\boldsymbol{x}, \boldsymbol{x}'). \tag{6}$$

The distance measure between depends on the microstructure representation and has to be chosen appropriately.

3. On the basis of the latent feature space, an extra autoencoder network is set up transforming $\boldsymbol{z} \in \mathbb{R}^M$ into an even lower-dimensional feature sub-space $\boldsymbol{s} \in \mathbb{R}^S$ with $S < M$ and transforming back to $\boldsymbol{z}' \in \mathbb{R}^M$ via

$$\boldsymbol{z}' = f_{\text{valid}}(\boldsymbol{z}, \boldsymbol{\theta}_{\text{valid}}) = f_{\text{valid}}(f_{\text{enc}}(\boldsymbol{x}, \boldsymbol{\theta}_{\text{enc}}), \boldsymbol{\theta}_{\text{valid}}). \tag{7}$$

The validity loss is defined by the mean squared error between z and z':

$$\mathscr{L}_{\text{valid}}(\boldsymbol{z}, \boldsymbol{z}') = \frac{1}{M} \sum_{i=1}^{M} (z_i - z_i')^2. \tag{8}$$

Distance preserving feature extraction using siamese neural networks

The above described MTL approach is used in combination with an optimizer that searches for candidate microstructures with desired properties in the latent feature space. However, our approach aims to identify a diverse set of microstructures with high diversity. For the diversity quantification a distance measure in the latent feature space is required. The MTL approach as defined above, is not able to preserve the distances of the original space in the latent feature space. In order to construct a distance preserving latent feature space, the MTL model is embedded in a siamese neural network (Bromley et al., 1993; Chicco, 2021), which we describe next.

Siamese neural networks consist of two identical networks, which share weights in the encoder part, see Fig. 4. Both networks embed different microstructures $\mathbf{x}_L$ and $\mathbf{x}_R$ as $\mathbf{z}_L$ and $\mathbf{z}_R$ in the latent feature space which is finally processed by two identical MTL networks. The distance preservation is enforced by defining a distance preservation loss $\mathscr{L}_{\text{pres}}$ that minimizes the difference between the distance of two different input microstructures in the original space $\text{dist}(\boldsymbol{x}_L, \boldsymbol{x}_R)$ and the corresponding distance in the latent feature space $\text{dist}(\boldsymbol{z}_L, \boldsymbol{z}_R)$, with $\boldsymbol{x}_L \neq \boldsymbol{x}_R$ (Utkin et al., 2017):

$$\mathscr{L}_{\text{pres}} = (\text{dist}(\boldsymbol{x}_L, \boldsymbol{x}_R) - \text{dist}(\boldsymbol{z}_L, \boldsymbol{z}_R))^2, \tag{9}$$

while $\text{dist}(\boldsymbol{x}_L, \boldsymbol{x}_R)$ and $\text{dist}(\boldsymbol{z}_L, \boldsymbol{z}_R)$ are not necessarily the same distance measures. Applying such loss terms leads to multi dimensional scaling, see Kruskal (1964) and Cox and Cox (2008). Using the distance preservation loss $\mathscr{L}_{\text{pres}}$, the MTL loss function, defined in Eq. 2, is extended by the
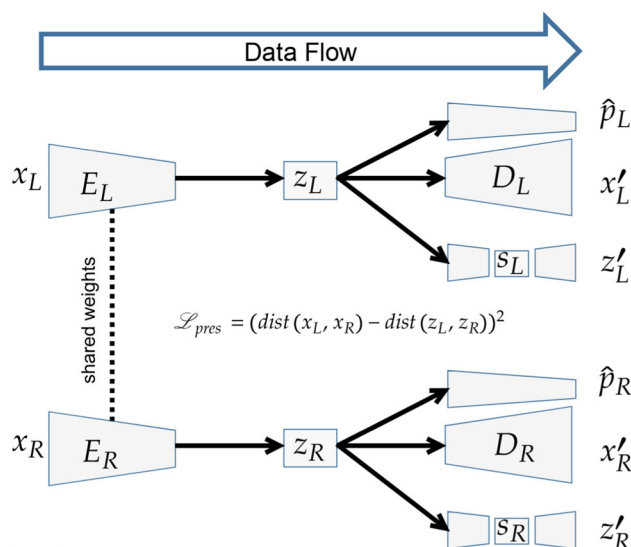

Fig. 4 Architecture of the SMTL approach. The dotted line between the encoders E_L and E_R indicates shared weights

weighted preservation loss $\mathscr{W}_{\text{pres}}\mathscr{L}_{\text{pres}}$ to

$$\begin{aligned} \mathscr{L}_{\text{SMTL}} = {}& \mathscr{W}_{\text{regr}}\mathscr{L}_{\text{regr}} + \mathscr{W}_{\text{recon}}\mathscr{L}_{\text{recon}} \\ & + \mathscr{W}_{\text{valid}}\mathscr{L}_{\text{valid}} + \mathscr{W}_{\text{pres}}\mathscr{L}_{\text{pres}} \\ & + \lambda R(\boldsymbol{\theta}). \end{aligned} \tag{10}$$

The SMTL approach delivers a function which can map a microstructure representation in the latent feature space on properties. Now, an optimizer can operate on a lower dimensional feature space to find microstructures with desired properties. The SMTL framework also allows to reconstruct the original represenation of microstructures, to asses the distances between them and to validate them in the latent feature space.

Microstructure optimizer

The microstructure optimization with respect to desired properties uses the distance preserving SMTL framework with the tasks microstructure–property mapping, validity-prediction and reconstruction. The optimization minimizes a loss function, which consists of the cost terms $\mathscr{C}_{\text{prop}}$, $\mathscr{C}_{\text{valid}}$ and $\mathscr{C}_{\text{divers}}$ and the corresponding weights $\mathscr{V}_{\text{prop}}$, $\mathscr{V}_{\text{valid}}$ and $\mathscr{V}_{\text{divers}}$:

$$\mathscr{F} = \mathscr{V}_{\text{prop}}\mathscr{C}_{\text{prop}} + \mathscr{V}_{\text{valid}}\mathscr{C}_{\text{valid}} + \mathscr{V}_{\text{divers}}(1 + \mathscr{C}_{\text{divers}}). \tag{11}$$

$\mathscr{C}_{\text{prop}}$, $\mathscr{C}_{\text{valid}}$ and $\mathscr{C}_{\text{divers}}$ denote the property, validity and diversity cost terms, respectively. While the property cost term drives the candidate microstructures to lie inside a specified target properties region, the validity cost aims that the optimizer operates inside the region of valid microstructures and the diversity cost ensures that candidate microstructures

differ from each other. To minimize the loss function we use genetic algorithms, which generate a population set of P candidate microstructures $\tilde{z}^*$ in the latent feature space in every iteration. The three cost terms are described in more detail in the following.

1. The property cost is defined by the mean squared error between the desired properties and the predicted properties from the SMTL regression model:

$$\mathscr{C}_{\text{prop}} = \frac{1}{N} \sum_{i=1}^{N} (\widetilde{\mathscr{C}}_{\text{prop},i})^2. \qquad (12)$$

If one of the predicted properties lies inside the target region, the cost $\widetilde{\mathscr{C}}_{\text{prop},i}$ equals 0. Otherwise, $\widetilde{\mathscr{C}}_{\text{prop},i}$ equals the minimum squared distance from the predicted properties to the target region borders.

2. The validity prediction is used to asses whether an identified candidate microstructure is likely to be represented by the sample data set. The validity cost is defined by

$$\mathscr{C}_{\text{valid}} = \max(\mathscr{A} - \xi_{\text{valid}}, 0), \qquad (13)$$

in which ξ_{valid} is a threshold to define the maximum tolerated reconstruction error for valid textures and $\mathscr{A}$ denotes the anomaly score

$$\mathscr{A} = \frac{1}{M} \sum_{i=1}^{M} (z_i^* - z_i^{*\prime})^2. \qquad (14)$$

3. The diversity cost is based on the sum of the distances between the candidate microstructure z^* in the latent feature space and every other microstructure in the population:

$$\mathscr{C}_{\text{divers}} = -\sum_{i=1}^{P} \text{dist}(z_i^*, z^*), \qquad (15)$$

in which for $\text{dist}(z_i^*, z^*)$ the same distance measure has to be used as for the latent feature vectors in Eq. 9.

Materials science fundamentals

Representation of crystallographic texture

Crystallographic texture is typically described by the orientation distribution function, which is defined by

$$f(g)\mathrm{d}g = \frac{\mathrm{d}V}{V}, \qquad (16)$$

for an orientation g (a point in $SO(3)$) and the volume $V(g)$ in $SO(3)$. The orientation distribution function $f(g)$ often

underlies specific symmetry conditions, for which various regions in $SO(3)$ are equivalent. Therefore, depending on the symmetries, orientations can be mapped into an elementary region of $SO(3)$, the so-called fundamental zone. The orientation distribution function on the basis of the orientations mapped into the fundamental zone is indistinguishable from the original orientation distribution function. Rolling textures, for example, underlie a cubic crystal and an orthorhombic sample symmetry, for which 96 elementary regions exist (Hansen et al., 1978).

A popular way to represent the orientation distribution function is by approximating it via generalized spherical harmonic functions (Bunge, 2013). Yet, as there is no straightforward way to measure the distance between two orientation distribution functions in terms of generalized spherical harmonics, we make use of the orientation histogram-based texture descriptor, which is introduced in Dornheim et al. (2021). Therefore, the cubic fundamental zone is discretized into a set O of J nearly uniform distributed orientations o_j. For each individual orientation g in a set of orientations G, a weight vector w_g is constructed via a soft-assignment approach

$$w_{\text{g}} = \begin{cases} \frac{\Phi(g,o_j)}{\sum_{o_i \in N_l} \Phi(g,o_i)}, & \text{if } o_j \in N_l \\ 0, & \text{else} \end{cases}, \qquad (17)$$

where N_l is the set of l nearest neighbor orientations of g in terms of the orientation distance Φ. The orientation distance between two orientations g and o is defined by

$$\Phi = \min \Phi(\overline{g}, \overline{o}). \qquad (18)$$

where $\overline{g}$ and $\overline{o}$ is from the set of all equivalent orientations of g and o in terms of cubic crystal symmetry. The orientation distance measure in $SO(3)$ is defined as

$$\Phi(q_{\text{g}}, q_{\text{o}}) = \min(||q_{\text{g}} - q_{\text{o}}||, ||q_{\text{g}} + q_{\text{o}}||), \qquad (19)$$

where q_{g} and q_{o} are the quaternion representations of the orientations g and o (Huynh, 2009).

On this basis, the weight vector for the orientation histogram $\boldsymbol{b}$ can be calculated by a volume average of the weight vectors of the individual orientations

$$\boldsymbol{b} = \frac{1}{V} \sum_{j=1}^{J} V(o_j) w_{o_j}. \qquad (20)$$

The distance between two orientation distribution functions can be measured via any kind of histogram-based distance measure, such as the Chi-Squared distance (Pele and Wer-

Table 2 Definition of the fibers of bcc rolling textures following Kocks et al. (1998)

Fiber	Location
α	From $\{001\}<110>$ to $\{111\}<1\bar{1}0>$, parallel to RD
γ	From $\{111\}<1\bar{1}0>$ to $\{111\}<112>$, parallel to ND
η	From $\{001\}<100>$ to $\{011\}<100>$, parallel to RD
ϵ	From $\{001\}<110>$ to $\{111\}<112>$, parallel to TD
β	From $\{112\}<1\bar{1}0>$ to $\{\bar{11}\ \bar{11}\ 8\}<4\ 4\ \bar{11}>$

man, 2010)

$$\chi^2(\boldsymbol{b}^{(1)}, \boldsymbol{b}^{(2)}) = \sum_{j=1}^{J} \frac{(b_j^{(1)} - b_j^{(2)})^2}{b_j^{(1)} + b_j^{(2)}}. \tag{21}$$

The set of nearly uniform distributed orientations O, needed for the histogram-based texture descriptor, can be generated using the algorithm described in Quey et al. (2018), which is implemented in the software *neper* (Quey et al., 2011). For the purpose of this study, we sample 512 nearly uniform distributed orientations over the cubic fundamental zone and chose a soft assignment of $l = 3$.

Crystallographic texture of steel sheets

After rolling body centered cubic (bcc) materials, typically so-called fiber textures are formed. Following (Ray et al., 1994), these textures are composed of the five fibers α, γ, η, ϵ, and β, which are defined in detail in Table 2. Among these fibers, the α and γ fiber are most prominent (Kocks et al., 1998), whereas the presence of the β fiber is only reported from theoretical predictions (Von Schlippenbach et al., 1986). To give an idea on how the fibers affect forming properties, we refer to Ray et al. (1994). Therein, it is found out that the γ fiber causes good deep drawability and the α fiber has a contrary effect.

In order to generate a data base of (artificial) rolling textures, in this work, a 25-parameter model is used, as it is proposed in Delannay et al. (1999) to describe steel sheet textures. The model is based on textures that are composed of the fibers α, γ, and η. As the η-fiber is not always present in steel sheet textures, we limit ourselves to textures that consist of an α and γ fiber. Therefore, 6 of the 25 parameters can be neglected.

The texture model describes the orientation distribution function as a set of weighted Gaussian distributions placed along the fibers. The model parameters D_i are listed in Table 3 and define the standard deviations and the mean values of the distributions based on the fiber thickness and the shifts from their ideal positions. Furthermore, the model parameters define the weights of the distributions among each other

based on the probability given by the orientation distribution function, what we will can fiber intensity in the following.

To construct the set of Gaussian distributions, the seven base distributions from Table 3 are placed at their ideal positions with respect to the shifts. Between these seven distributions, further distributions are placed with a distance of about 3° to each other, leading to overall 41 Gaussians. Their weights w_i and the values for the standard deviation σ_i and mean value μ_i are interpolated linearly based on the values of the two neighboring base distributions. This yields a set of Gaussian distributions $\mathcal{N}_1(\mu_1, \sigma_1), ..., \mathcal{N}_{41}(\mu_{41}, \sigma_{41})$. The orientation distribution function $f(g)$ is defined by the normalized sum of this set:

$$f(g) = \frac{1}{\sum_i w_i} \sum_{i=1}^{n} w_i \mathcal{N}_i(\mu_i, \sigma_i). \tag{22}$$

Based on this definition, discrete orientations can be sampled. In the following, we denote the set of orientations as G. As $f(g)$ is defined in the cubic-orthorhombic fundamental zone, it is necessary to add the equivalent orientations regarding the orthorhombic sample symmetry to the set of discrete orientations. This is done by applying rotation operations g_s on each orientation g_i in G

$$g_i^{\text{equiv}} = g_s\, g_i. \tag{23}$$

The rotation operations g_s for orthorhombic sample symmetry can be found in Hansen et al. (1978).

Material model

The sheet metal properties which we focus on in this study are the Young's moduli and the r-values at 0, 45 and 90 degree to rolling direction. In this study, the properties are calculated by applying uniaxial tension on a crystal plasticity-based material model. As time efficiency is essential for the generation of data, a material model of Taylor-type is implemented, as it is described in Dornheim et al. (2021).

The Taylor-type material model is based on the volume averaged stress of a set of n crystals (Kalidindi et al., 1992):

$$\overline{\boldsymbol{T}} = \frac{1}{V} \sum_{i=1}^{n} \boldsymbol{T}^{(i)} V^{(i)}. \tag{24}$$

In the above equation, $\boldsymbol{T}$ denotes the Cauchy stress tensor, which can be derived by the stress tensor in the intermediate configuration, given by

$$\boldsymbol{T}^* = \frac{1}{2}\, \mathbb{C} : (\boldsymbol{F}_e^T \cdot \boldsymbol{F}_e - \boldsymbol{I}), \tag{25}$$

Table 3 Definition of the parameters D_i of the texture model, cf. Delannay et al. (1999)

	Ideal pos.	Intens.	Std φ_2	Std φ_1	Shift φ_1	Shift ϕ	Shift φ_2
a_1	0, 0, 45	D_1	D_7	D_7	0	0	0
a_2	0, 30, 45	D_2	D_8	D_{13}	0	D_{15}	0
a_3	0, 55, 45	D_3	D_9	D_9	0	D_{16}	0
a_4	0, 70, 45	D_4	D_{10}	D_{10}	0	D_{17}	0
a_5	0, 90, 45	D_4	D_{10}	D_{10}	0	0	0
g_2	15, 55, 45	D_5	D_{11}	D_{11}	D_{14}	0	0
g_3	30, 55, 45	D_6	D_{12}	D_{12}	0	D_{18}	D_{19}

The ideal position is given in Bunge Euler angles [°]

with the second order identity tensor I and the fourth order elastic stiffness tensor $\mathbb{C}$. The elastic constants C_{11}, C_{12} and C_{44} are set to 218.37, 131.13 and 105.34 GPa, respectively (Eghtesad and Knezevic, 2020). F_e is the elastic part of the deformation gradient F and can be calculated by a multiplicative decomposition

$$F = F_\mathrm{e} \cdot F_\mathrm{p}. \tag{26}$$

The intermediate stress tensor can be converted into Cauchy stress using the relation

$$T^* = F_\mathrm{e}^{-1} \cdot (\det(F_\mathrm{e})\, T) \cdot F_\mathrm{e}^{-\top}. \tag{27}$$

To describe the evolution of the plastic deformation, the plastic part of the velocity gradient L_p is considered by

$$L_\mathrm{p} = \dot{F}_\mathrm{p} \cdot F_\mathrm{p}^{-1}, \tag{28}$$

and the flow rule (Rice, 1971)

$$L_\mathrm{p} = \sum_\eta \dot{\gamma}^{(\eta)} m^{(\eta)} \otimes n^{(\eta)}, \tag{29}$$

where $\dot{\gamma}^{(\eta)}$ denotes the shear rates on the active slip systems η, defined by the slip plane normal $n^{(\eta)}$ and the slip direction $m^{(\eta)}$. For bcc materials, the slip system families in terms of the Miller index are {110}<111>, {112}<111>, and {123}<111>, while the latter is neglected due to simplicity.

The shear rates are defined by a phenomenological power-law (Asaro & Needleman, 1985):

$$\dot{\gamma}^{(\eta)} = \dot{\gamma}_0 \left| \frac{\tau^{(\eta)}}{r^{(\eta)}} \right|^{1/m} \mathrm{sign}(\tau^{(\eta)}), \tag{30}$$

where $r^{(\eta)}$ is the slip system resistance, $\dot{\gamma}_0$ the reference shear rate and m the shear rate sensitivity. Here, $\dot{\gamma}_0$ and m are set to 0.001 sec^{-1} and 0.0125, respectively (Pagenkopf et al., 2016). Following Schmid's law, the resolved shear stress on

slip system $\tau^{(\eta)}$ is given by

$$\tau^{(\eta)} = ((F_\mathrm{e}^T \cdot F_\mathrm{e}) \cdot T^*) : (m^{(\eta)} \otimes n^{(\eta)}), \tag{31}$$

and the evolution of the slip system resistance is defined by

$$\dot{r}^{(\eta)} = \frac{\mathrm{d}\hat{\tau}^{(\eta)}}{\mathrm{d}\Gamma} \sum_\xi q_{\eta\xi} |\dot{\gamma}^{(\xi)}|. \tag{32}$$

The matrix $q_{\eta\xi}$ describes the ratio between self and latent hardening. It consists of diagonal elements equal to 1.0 and off-diagonal elements q_1 and q_2, cf. Baiker et al. (2014). Both, q_1 and q_2, are set to 1.4 (Asaro and Needleman, 1985). Further, the hardening behavior is realized by an extended Voce-type model (Tome et al., 1984):

$$\hat{\tau}^{(\eta)} = \tau_0 + (\tau_1 + \vartheta_1 \Gamma)(1 - e^{-\Gamma \vartheta_0/\tau_1}). \tag{33}$$

The material dependent parameters are calibrated to DC04 steel[1] and are $\tau_0 = 94.9$ MPa, $\tau_1 = 50$ MPa, $\vartheta_0 = 258$ MPa and $\vartheta_1 = 32.8$ MPa (Pagenkopf, 2019). The accumulated plastic shear is defined by

$$\Gamma = \int_0^t \sum_\eta \left| \dot{\gamma}^{(\eta)} \right| \mathrm{d}t. \tag{34}$$

Although material parameters for DC04 steel are used in this study, it is to remark that the described Taylor-type crystal plasticity model and the texture generation approach can be applied to any kind of metallic material with bcc crystal structure.

Results

Texture-property data set

For training, 50000 sets of 2000 discrete orientations are sampled via Latin hypercube design (McKay et al., 1979), based

---

[1] Experiments performed at IUL Dortmund during DFG project Graduate School 1483 (Pagenkopf, 2019).

Table 4 Parameter ranges for D_i

Intensity	D_1	D_2	D_3	D_4	D_5	D_6	
Min	1/6	1/6	1/6	0	1/6	1/6	
Max	1/3	2/3	2/3	1/3	1	1	
Std	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}
Min	5/3	5/3	5/3	5/3	5/3	5/3	5/3
Max	45/3	35/3	35/3	30/3	30/3	30/3	35/3
Shift	D_{14}	D_{15}	D_{16}	D_{17}	D_{18}	D_{19}	
Min	−5	−10	−10	−5	−10	−10	
Max	10	10	5	10	10	10	
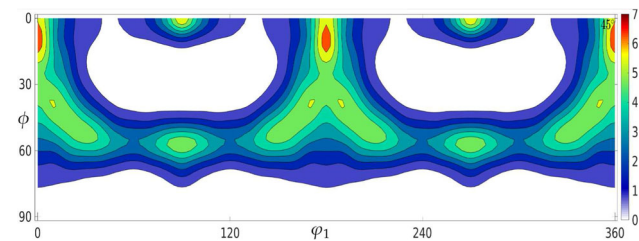
Fig. 5 $\varphi_2 = 45°$ section of the orientation distribution function to visualize the γ-fiber of the colled rolled DC04 steel texture

Fig. 6 One twin part of the SMTL model with the annotation of the dimension size of the layers. *Fc* denotes fully-connected layers and *tanh* denotes hyperbolic tangent activation function

Table 5 Used hyperparamters

Hyperparameter	Value
Optimizer	Adam (Kingma and Ba, 2015)
Learning rate	0.001
Weight decay	1e-6
Batch size	256

on Eq. 22. In order to have an independent test set, further 10000 sets are generated randomly. The ranges inside which the parameters of the texture model vary are adjusted manually such that typical bcc rolling textures found in literature Das (2017), Hölscher et al. (1991), Inagaki and Suda (1972), Kestens and Pirgazi (2016), Klinkenberg et al. (1992), Kocks et al. (1998), and Pagenkopf et al. (2016) are covered. The parameter ranges are listed in Table 4. In addition, to evaluate the anomaly detection, a set of artificial textures is needed, which slightly differ from the generated rolling textures. For this purpose, 10000 anomalies are generated by shifting the α-fiber (i.e. the ideal position of a_1, a_2, a_4 and a_5) about 20 degrees in φ_1-direction.

Moreover, we validate the texture-property mapping and the validity-prediction on experimental data. For this purpose, an experimentally measured texture of cold rolled DC04 steel from Schreijäg (2012) is used. Based on this measurement, an orientation distribution function is approximated via the MATLAB toolbox mtex (Bachmann et al., 2010), rotated into its symmetry axis assuming orthorhombic sample symmetry and mirrored. To visualize the α- and γ-fiber of the orientation distribution, an intersection plot of the Euler space at $\varphi_2 = 45°$ is depicted in Fig. 5.

For the generated textures in the training and test set, the corresponding Young's moduli and R-values in 0, 45, and 90 degree to rolling direction are determined using the Taylor-type crystal plasticity model described in "Material model" section. Both quantities, Young's modulus and especially R-values, are highly affected by the crystallographic texture,
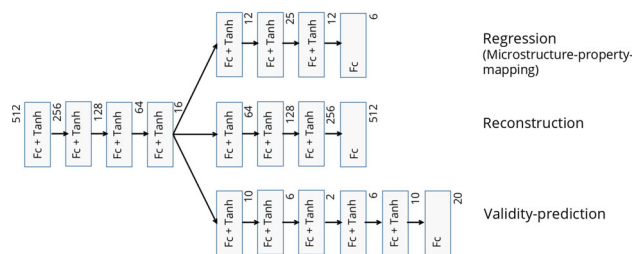
which is why these are chosen exemplary for the purpose of this study.

Validation of SMTL

In this study, the individual tasks of the SMTL model are realized via feedforward neural networks with *tanh* activation functions to obtain features between −1 and +1 in the latent feature space. The SMTL model is implemented based on the Python TensorFlow API (Abadi et al., 2015). The base network of the siamese architecture is illustrated in Fig. 6. The Glorot Normal method (Glorot & Bengio, 2010) is used for weight initialization. In order to adjust the hyperparameters, a random search method (Bergstra & Bengio, 2012) is applied using 5-fold cross-validation.

The best model configuration that was found is shown in Table 5. We use the Chi-Squared distance introduced in Eq. 21 as distance measure in the input space. In the latent feature space, we use the sum of squared errors (SSE) between two vectors z_1 and z_2 as distance measure

$$\text{SSE}(z_1, z_2) = \sum_{i=1}^{M} (z_{1,i} - z_{2,i})^2. \tag{35}$$

The SMTL model is trained for 200 epochs, while the best intermediate result of the test set is retained, which can be interpreted as early stopping (Prechelt, 1998). Before the model training is executed, the loss terms are scaled to values between 0 and 1 in order to make them comparable. The following weights for the scaled loss terms were based on hyper parameter optimization: $\mathscr{W}_{\text{regr}} = 0.05$, $\mathscr{W}_{\text{recon}} = 0.05$, $\mathscr{W}_{\text{valid}} = 0.05$ and $\mathscr{W}_{\text{pres}} = 0.85$.

Table 6 Results for varying numbers of latent features (LF) of the texture-property mapping and the distance preservation applied to the artificially generated textures and experimentally measured texture

LF	Artificial textures			Experimental texture	
	Regr. MAE_E	Regr. MAE_r	Pres. R^2	Regr. MAE_E	Regr. MAE_r
20	153	0.03	98.2	596	0.04
18	183	0.03	98.0	773	0.11
16	162	0.03	97.8	907	0.05
14	193	0.03	97.0	1282	0.07
12	215	0.04	95.4	1468	0.13
10	238	0.05	92.2	1463	0.13
8	335	0.06	85.7	1575	0.17
6	390	0.07	72.4	1554	0.15
4	664	0.10	34.2	2768	0.12

Regr.(ession error) MAE_E is given in [MPa], Regr.(ession error) MAE_r in [-] and Pres.(erve quality) R^2 in [%]

The results for the texture-property mapping and the distance preservation are shown in Table 6, in which the regression errors MAE_E and MAE_r denote the mean absolute error between the true and predicted Young's moduli and r-values depending on the dimension of the latent feature space z. The quality of the distance preservation is measured by the coefficient of determination R^2, between the distances of two input textures and their corresponding latent feature vectors

$$R^2(\chi^2(\mathbf{x}_L, \mathbf{x}_R), SSE(\mathbf{z}_L, \mathbf{z}_R)). \tag{36}$$

It is shown that texture-property mappings with an adequate prediction quality can be achieved by extensively reducing the dimensionality of the latent feature space. However, regarding the distance preservation quality, a lower bound of at least 10 latent features can be identified, below which the distance preservation is unsatisfactory. Additionally, the texture-property mapping is evaluated on the experimentally measured texture and the corresponding properties. The results are listed in Table 6. It can be seen that a satisfactory prediction quality (Regr. $MAE_E \leq 1000$ MPa and Regr. $MAE_r \leq 0.1$) can only be achieved for at least 16 latent features.

On the basis of this 16-dimensional feature space, the validity-prediction is evaluated. The anomaly scores for the textures in the test set and for the artificially generated anomalies are shown in Fig. 7. It can be seen, that the anomalies can be separated in a sufficient manner from the textures in the test set.

Rolling texture identification

To validate the texture identification, we define two target regions in property space, see Fig. 8. The first one is defined by the properties of the experimentally measured texture, which lies in a sparsely populated region and is labeled as *Target Region 1*. As a consequence of its location in the
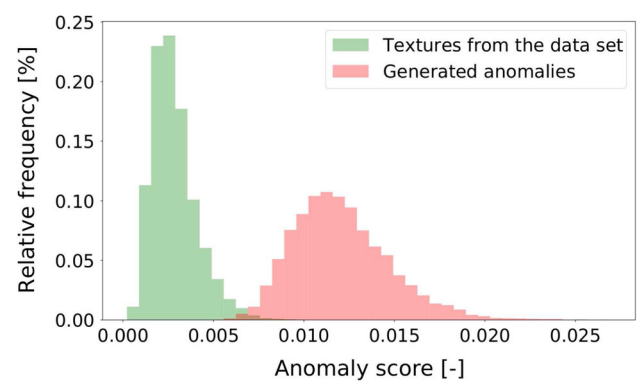

Fig. 7 Histograms of the anomaly scores for the data from the test set and the set of artificially generated anomalies. The anomaly scores are based on the model that uses 16 latent features

sparsely populated region, the anomaly score of this texture is 0.0099 and lies in the transition zone shiftet towards the generated anomalies (cf. Fig. 7). It is of interest if the optimizer is generally able to find a whole set of microstructures with properties in this region. The second target region represents a densely populated region located near the center of the properties point cloud and is labeled as *Target Region 2*. The center of each target region is listed in Table 7. The target regions are defined by adding a tolerance of ± 1000 MPa to the Young's moduli and ± 0.10 to the r-values, yielding a sufficiently small properties window from an engineering point of view. As a baseline, we collect all data points from the training set, that lie inside the target regions. In *Target Region 1* only two textures can be found, whereas in *Target Region 2* 13 textures can be found.

To identify a diverse set of textures, we use the optimization algorithm JADE (Zhang and Sanderson, 2009), which is an extension of the differential evolution algorithm (Storn and Price, 1997). Before starting the optimization via JADE, an initial population has to be selected: Therefore, 100 textures are sampled from the test set, which are approximately
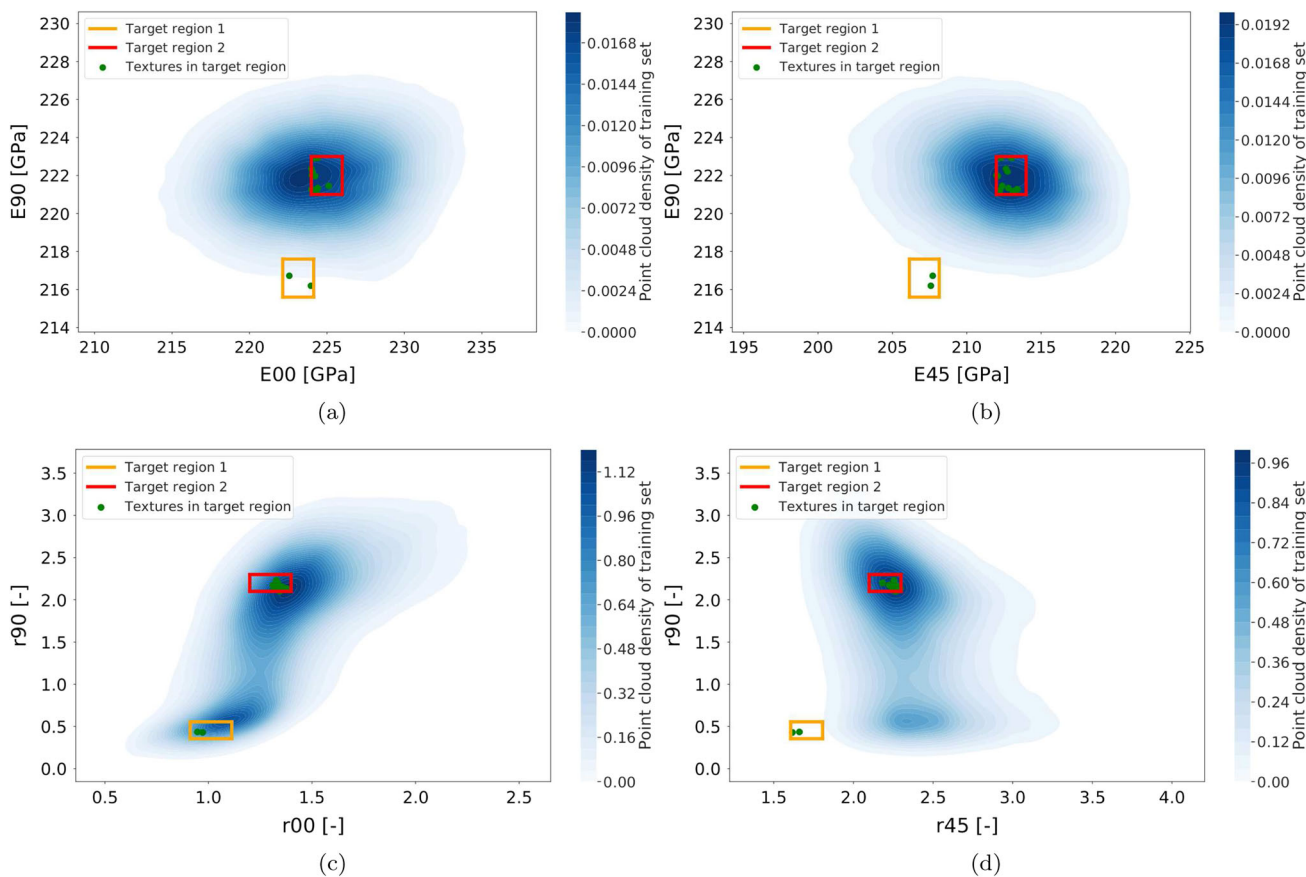
Fig. 8 Density of the training set projected on different planes of the property space: the Young's modulus at **a** 0 vs. 90 degree and **b** 45 vs. 90 degree for the r-values at **c** 0 vs. 90 degree and **d** 45 vs. 90 to rolling direction. The orange and red squares mark the projections of *Target Region 1* and *Target Region 2*, respectively. The green dots show the projected samples from the training set that lie inside the target region

Table 7 Center points of the two target regions (TR)

TR	E_{00}	E_{45}	E_{90}	r_{00}	r_{45}	r_{90}
1	223145	207148	216599	1.01	1.7	0.4
2	223000	213000	222000	1.3	2.2	2.2

The Young's moduli E are given in MPa, the r-values r in $[-]$

uniformly distributed over the property space. For the cost function, defined in Eq. 11, we use the weights $\mathcal{V}_{prop} = 0.90$, $\mathcal{V}_{valid} = 0.03$ and $\mathcal{V}_{divers} = 0.07$ and scale C_{props} and C_{divers} to values between 0 and 1 based on the selected 100 initial textures. The threshold ξ_{valid} is set to 0.01 based on the maximum anomaly score in the data set, cf. Fig. 7. The optimization is performed for 300 iterations with a fixed population size of 100. During the optimization, all valid textures that fulfill the target properties are collected, according to the texture-property mapping. Based on the results from the previous section, we use the trained SMTL model with a 16-dimensional latent feature space.

Target region 1

Our approach is able to find a diverse set of textures that meet the property requirements of *Target Region 1*, according to the texture-property mapping. Figure 9 depicts the mutual distances in the latent feature space between all the found textures and between the two baseline textures. It is shown, that the set of identified textures contains 221 diverse textures in contrary to only two in the baseline set. In order to compare the results to the experimentally measured texture, the closest texture to the center point of *Target Region 1* is depicted in Fig. 10 as a section through the Euler space at $\varphi_2 = 45°$. By comparing the two textures, it can be seen that they are roughly the same in terms of the magnitude of the intensities and the shape of the α- and γ-fibers. However, they also show differences in terms of smoothness and the location of the intensity peaks.
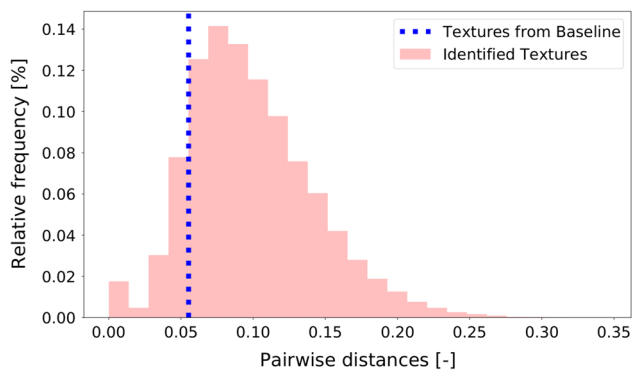
Fig. 9 Histogram of pairwise SSE distances of the set of identified textures and the baseline set for *Target Region 1*. The distance between the two textures from the baseline set is indicated by the dashed line
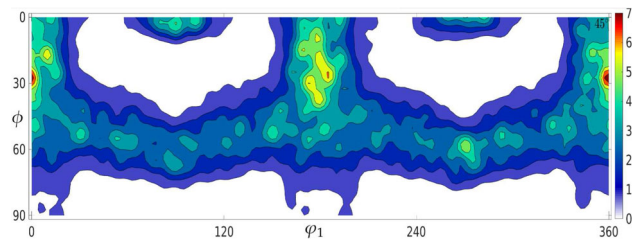

Fig. 10 Texture that yields properties which are closest to the center of *Target Region 1*. The plot shows the $\varphi_2 = 45°$ section of the orientation distribution function
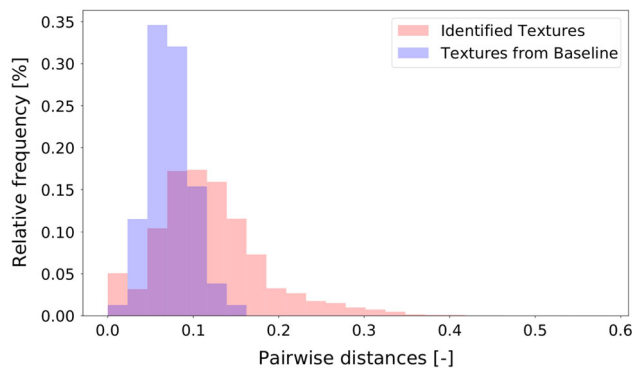

Fig. 11 Histogram of mutual distances of the set of identified textures and the baseline set for *Target Region 2*

Target region 2

Compared to *Target Region 1*, an even more diverse set of 1315 textures can be identified for *Target Region 2*, which can be seen in the histogram of the mutual distances in Fig. 11. To get an idea of the differences between the textures, two exemplary textures are plotted in Fig. 12 as a section through the Euler space at $\varphi_2 = 45°$. It can be seen that the α- and γ-fiber of both textures differ significantly in terms of intensity. However, the locations of the intensity peaks and the thickness of the α- and γ-fiber are similar.
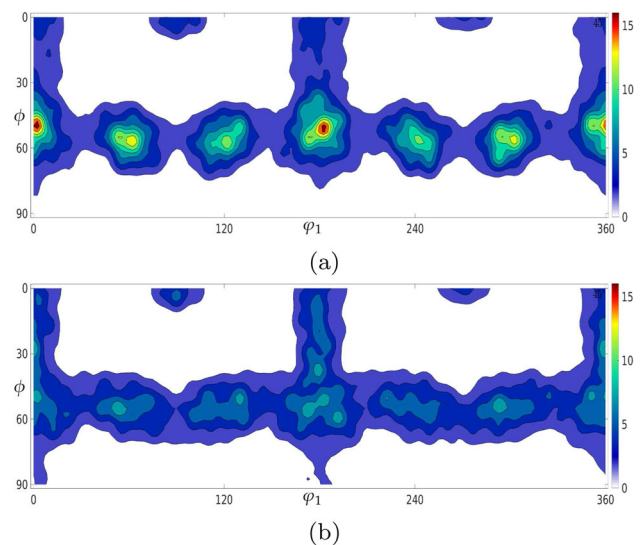

Fig. 12 Two exemplary textures from the set of identified textures. Both plots show $\varphi_2 = 45°$ sections of the respective orientation distribution functions

Discussion

The results presented in "Validation of SMTL" section show that the two tasks texture-property mapping and validity-prediction are solved by the SMTL model. To achieve a sufficient prediction quality for both tasks in the test set as well as for the experimentally measured texture, a minimum dimensionality of the latent feature space is needed. Here, also the dimensionality requirements of the siamese distance preservation goal has to be considered. 16 latent features were found to be sufficient for our example task regarding the texture of cold rolled bcc steel sheets.

However, the prediction error for the experimentally measured texture is higher than for the test set using the same latent feature space dimensionality. This can be explained by the fact that the corresponding property is in a texture space region with low sampling density and the model therefore is not well supported by data. This results also in an instability of the model quality depending on the dimensionality of the latent feature space in this region. This instability can be seen by studying the r-value in Table 6. By choosing the latent feature space size of 16, also the results for the experimentally measured texture are satisfactory, especially keeping in mind that the experimentally measured texture differs naturally from the artificially generated data and additionally lies in a sparsely sampled region, cf. *Target Region 1* in Fig. 8.

Due to the sparsity of *Target Region 1*, the identification of textures in this region is challenging. Nevertheless, the optimization approach is able to identify a set of textures that contains more diverse individuals compared to the two baseline textures from the training set. Regarding the identified texture, which is closest to the experimentally measured

texture in terms of properties, one can see that they are also similar in terms of crystallographic texture, what basically proofs the concept of our approach.

The most obvious difference between both textures is smoothness. The irregular distribution of intensity peaks of the identified texture is due to the resolution of the histogram-based texture descriptor. Also, the orthorhombic sample symmetry is not represented locally. However, by increasing the resolution, these two issues can be solved. Furthermore, a higher resolution of the descriptor decreases the descriptor error, which reflects the deviation between the properties of the original texture and the properties of the texture described by the descriptor. However, the choice of resolution is a trade-off between accuracy and descriptor complexity. Generally, with the use of the SMTL model and the incorporated feature extraction, the resolution is limited only by computational power.

Compared to *Target Region 1*, the identification task for *Target Region 2* seems to be less challenging as the target region is located in a densely sampled region. However, as there already exists a proper set of diverse textures in the baseline, the main challenge is to outperform the baseline set in terms of diversity. Figure 11 shows that the materials design problem (the identification of multiple equivalent microstructures/ textures) is accomplished by the optimization approach. This is exemplary shown when comparing two of the identified textures in Fig. 12 with each other: similar properties can be reached by different microstructures. The identification of such a highly diverse set of microstructures with similar properties is an important precondition to construct robust optimizing process control algorithms, which need to choose among multiple optimal paths leading to desired properties.

Summary and outlook

In this work, we present an approach to solve materials design problems. The approach is based on an optimization strategy that incorporates machine learning models for mapping microstructures to properties and for assessing the validity of input microstructures in the sense of the likeliness with the underlying data. To model these tasks, we use a siamese multi-task learning (SMTL) neural network model. Furthermore, we incorporate feature extraction in order to transform input microstructures to a lower dimensional latent feature space, in which an optimizer (identifying microstructures with dedicated properties) can operate efficiently.

By training the SMTL model with a dedicated loss function term, we are able to preserve the distances between microstructures in the original input space also in the latent feature space. The distance preservation allows to directly assess the diversity of the solution set (found by the opti-

mizer) directly in the latent feature space and therefore enables optimizers to efficiently identify sets of diverse microstructures. By applying the approach to crystallographic texture optimization, we show the ability to identify diverse sets of textures that lie within given properties bounds. Such sets of textures form the input of optimal processing control approaches like in Dornheim et al. (2021).

In the present work, we applied our approach on data from mean-field simulations. The next step is to apply the approach on spatially resolved data from full-field simulations. The proposed methods can be easily extended for this task by modifying the encoder part of the SMTL model. However, the problem arises that typically fewer data can be generated via full-field simulations. Nevertheless, such sparse high quality data can be used to support the modeling with lower quality data. Concepts to incorporate multi-fidelity data (Batra, 2021) in our SMTL model will be considered in the future.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro,C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., & Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. White paper.

Adams, B. L., Henrie, A., Henrie, B., Lyon, M., Kalidindi, S., & Garmestani, H. (2001). Microstructure-sensitive design of a compliant beam. *Journal of the Mechanics and Physics of Solids, 49*(8), 1639–1663.

Asaro, R. J., & Needleman, A. (1985). Overview no. 42 texture development and strain hardening in rate dependent polycrystals. *Acta Metallurgica, 33*(6), 923–953.

Bachmann, F., Hielscher, R., & Schaeben, H. (2010). Texture analysis with mtex - free and open source software toolbox. *Solid State Phenomena, 160*, 63–68.

Baiker, M., Helm, D., & Butz, A. (2014). Determination of mechanical properties of polycrystals by using crystal plasticity and numerical homogenization schemes. *Steel Research International, 85*(6), 988–998.

Batra, R. (2021). Accurate machine learning in materials science facilitated by using diverse data sources. *Nature*, 589.

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research, 13*(10), 281–305.

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., & Shah, R. (1993). Signature verification using a Siamese time delay neural network. *Advances in Neural Information Processing Systems, 6*, 737–744.

Bunge, H.-J. (2013). *Texture analysis in materials science: Mathematical methods*. Burlington: Elsevier Science.

Caruana, R. (1997). Multitask learning. *Machine Learning, 28*(1), 41–75.

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

Chalapathy, R., & Chawla, S. (2019). Deep learning for anomaly detection: A survey. arXiv:1901.03407

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys (CSUR), 41*, 3.

Cheng, Z., Wang, S., Zhang, P., Wang, S., Liu, X., & Zhu, E. (2021). Improved autoencoder for unsupervised anomaly detection. *International Journal of Intelligent Systems, 36*, 7103–7125.

Chicco, D. (2021). Siamese neural networks: An overview. *Artificial Neural Networks*, 73–94.

Cox, M. A., & Cox, T. F. (2008). Multidimensional scaling. In *Handbook of data visualization* (pp. 315–347). Springer.

Das, A. (2017). Calculation of crystallographic texture of bcc steels during cold rolling. *Journal of Materials Engineering and Performance, 26*(6), 2708–2720.

Delannay, L., Van Houtte, P., & Van Bael, A. (1999). New parameter model for texture description in steel sheets. *Texture, Stress, and Microstructure, 31*(3), 151–175.

Dornheim, J., Morand, L., Zeitvogel, S., Iraki, T., Link, N., & Helm, D. (2021). Deep reinforcement learning methods for structure-guided processing path optimization. *Journal of Intelligent Manufacturing*

Eghtesad, A., & Knezevic, M. (2020). High-performance full-field crystal plasticity with dislocation-based hardening and slip system back-stress laws: Application to modeling deformation of dual-phase steels. *Journal of the Mechanics and Physics of Solids, 134*, 103750.

Fullwood, D. T., Niezgoda, S. R., Adams, B. L., & Kalidindi, S. R. (2010). Microstructure sensitive design for performance optimization. *Progress in Materials Science, 55*(6), 477–562.

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the 13th international conference on artificial intelligence and statistics* (pp. 249–256). JMLR Workshop and Conference Proceedings

Goldberg, D. (1991). Real-coded genetic algorithms, virtual alphabets and blocking. *Complex Systems,5*.

Gupta, A., Cecen, A., Goyal, S., Singh, A. K., & Kalidindi, S. R. (2015). Structure-property linkages using a data science approach: Application to a non-metallic inclusion/steel composite system. *Acta Materialia, 91*, 239–254.

Hansen, J., Pospiech, J., & Lücke, K. (1978). *Tables for texture analysis of cubic crystals*. Springer.

Herrera, F., Lozano, M., & Verdegay, J. L. (1998). Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis. *Artificial Intelligence Review, 12*(4), 265–319.

Hinton, G. E. (1987). Learning translation invariant recognition in a massively parallel networks. In *International conference on parallel architectures and languages Europe* (pp. 1–13). Springer.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science, 313*(5786), 504–507.

Hölscher, M., Raabe, D., & Lücke, K. (1991). Rolling and recrystallization textures of bcc steels. *Steel Research, 62*(12), 567–575.

Huynh, D. Q. (2009). Metrics for 3D rotations: Comparison and analysis. *Journal of Mathematical Imaging and Vision, 35*(2), 155–164.

Inagaki, H., & Suda, T. (1972). The development of rolling textures in low-carbon steels. *Texture, Stress, and Microstructure, 1*(2), 129–140.

Jung, J., Yoon, J. I., Park, H. K., Jo, H., & Kim, H. S. (2020). Microstructure design using machine learning generated low dimensional and continuous design space. *Materialia, 11*, 100690.

Jung, J., Yoon, J. I., Park, H. K., Kim, J. Y., & Kim, H. S. (2019). An efficient machine learning approach to establish structure-property linkages. *Computational Materials Science, 156*, 17–25.

Jung, J., Yoon, J. I., Park, S.-J., Kang, J.-Y., Kim, G. L., Song, Y. H., Park, S. T., Oh, K. W., & Kim, H. S. (2019). Modelling feasibility constraints for materials design: Application to inverse crystallographic texture problem. *Computational Materials Science, 156*, 361–367.

Kalidindi, S. R., Bronkhorst, C. A., & Anand, L. (1992). Crystallographic texture evolution in bulk deformation processing of FCC metals. *Journal of the Mechanics and Physics of Solids, 40*(3), 537–569.

Kalidindi, S. R., Houskamp, J. R., Lyons, M., & Adams, B. L. (2004). Microstructure sensitive design of an orthotropic plate subjected to tensile load. *International Journal of Plasticity, 20*(8–9), 1561–1575.

Kamijyo, R., Ishii, A., Coppieters, S., & Yamanaka, A. (2022). Bayesian texture optimization using deep neural network-based numerical material test. *International Journal of Mechanical Sciences, 223*, 107285.

Kestens, L., & Pirgazi, H. (2016). Texture formation in metal alloys with cubic crystal structures. *Materials Science and Technology, 32*(13), 1303–1315.

Kingma, D. P. & Ba, J. (2015). Adam: A method for stochastic optimization. In: *3rd international conference on learning representations*

Klinkenberg, C., Raabe, D., & Lücke, K. (1992). Influence of volume fraction and dispersion rate of grain-boundary cementite on the cold-rolling textures of low-carbon steel. *Steel Research, 63*(6), 263–269.

Kocks, U. F., Tomé, C. N., & Wenk, H.-R. (1998). *Texture and anisotropy: Preferred orientations in polycrystals and their effect on materials properties*. Cambridge University Press.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems, 25*, 1106–1114.

Krogh, A., & Hertz, J. A. (1991). A simple weight decay can improve generalization. *Advances in Neural Information Processing Systems, 4*, 950–995.

Kruskal, J. (1964). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika, 29*, 1–27.

Kuroda, M., & Ikawa, S. (2004). Texture optimization of rolled aluminum alloy sheets using a genetic algorithm. *Materials Science and Engineering: A, 385*(1–2), 235–244.

Kwon, G., Prabhushankar, M., Temel, D., & AlRegib, G. (2020). Back-propagated gradient representations for anomaly detection. In: *European conference on computer vision*

Liu, R., Kumar, A., Chen, Z., Agrawal, A., Sundararaghavan, V., & Choudhary, A. (2015). A predictive machine learning approach for microstructure optimization and materials design. *Scientific Reports, 5*(1), 1–12.

Mann, A., & Kalidindi, S. R. (2022). Development of a robust CNN model for capturing microstructure-property linkages and building property closures supporting material design. In *Frontiers in materials*

McDowell, D. L. (2007). Simulation-assisted materials design for the concurrent design of materials and products. *JOM, 59*(9), 21–25.

McKay, M. D., Beckman, R. J., & Conover, W. J. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics, 21*(2), 239.

Morand, L., Iraki, T., Dornheim, J., Pagenkopf, J., & Helm, D. (2021). Artificially generated crystallographic textures of steel sheets and their corresponding properties calculated by a Taylor-type crystal plasticity model. Retrieved from https://fordatis.fraunhofer.de/handle/fordatis/204

Morand, L., Link, N., Iraki, T., Dornheim, J., & Helm, D. (2022). Efficient exploration of microstructure-property spaces via active learning. *Frontiers in Materials, 8*, 824441. https://doi.org/10.3389/fmats

Olson, G. B. (1997). Computational design of hierarchically structured materials. *Science, 277*(5330), 1237–1242.

Pagenkopf, J. (2019). *Bestimmung der Plastischen Anisotropie von Blechwerkstoffen durch ortsaufgelöste Simulationen auf Gefügeebene.* PhD thesis, Fakultät für Maschinenbau des Karlsruher Instituts für Technologie (KIT).

Pagenkopf, J., Butz, A., Wenk, M., & Helm, D. (2016). Virtual testing of dual-phase steels: Effect of martensite morphology on plastic flow behavior. *Materials Science and Engineering A, 674*, 672–686.

Panchal, J. H., Kalidindi, S. R., & McDowell, D. L. (2013). Key computational modeling issues in integrated computational materials engineering. *Computer-Aided Design, 45*(1), 4–25.

Paul, A., Acar, P., Liao, W.-K., Choudhary, A., Sundararaghavan, V., & Agrawal, A. (2019). Microstructure optimization with constrained design objectives using machine learning-based feedback-aware data-generation. *Computational Materials Science, 160*, 334–351.

Paulson, N. H., Priddy, M. W., McDowell, D. L., & Kalidindi, S. R. (2017). Reduced-order structure-property linkages for polycrystalline microstructures based on 2-point statistics. *Acta Materialia, 129*, 428–438.

Pele, O., & Werman, M. (2010). The quadratic-chi histogram distance family. In *European conference on computer vision* (pp. 749–762). Springer.

Prechelt, L. (1998). Early stopping-but when?. In *Neural networks: Tricks of the trade* (pp. 55–69). Springer

Quey, R., Dawson, P., & Barbe, F. (2011). Large-scale 3d random polycrystals for the finite element method: Generation, meshing and remeshing. *Computer Methods in Applied Mechanics and Engineering, 200*(17–20), 1729–1745.

Quey, R., Villani, A., & Maurice, C. (2018). Nearly uniform sampling of crystal orientations. *Journal of Applied Crystallography, 51*(4), 1162–1173.

Ramprasad, R., Batra, R., Pilania, G., Mannodi-Kanakkithodi, A., & Kim, C. (2017). Machine learning in materials informatics: Recent applications and prospects. *NPJ Computational Materials, 3*(1), 1–13.

Ray, R., Jonas, J. J., & Hook, R. (1994). Cold rolling and annealing textures in low carbon and extra low carbon steels. *International Materials Reviews, 39*(4), 129–172.

Rice, J. R. (1971). Inelastic constitutive relations for solids: An internal-variable theory and its application to metal plasticity. *Journal of the Mechanics and Physics of Solids, 19*(6), 433–455.

Ruff, L., Görnitz, N., Deecke, L., Siddiqui, S. A., Vandermeulen, R. A., Binder, A., Müller, E., & Kloft, M. (2018). Deep one-class classification. In *International Conference on Machine Learning.,*

Ruff, L., Kauffmann, J. R., Vandermeulen, R. A., Montavon, G., Samek, W., Kloft, M., Dietterich, T. G., & Müller, K.-R. (2021). A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE, 109*(5), 756–795.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature, 323*, 533–536.

Sakurada, M., & Yairi, T. (2014). Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis* (pp. 4–11).

Schölkopf, B., Platt, J. C., Shawe-Taylor, J. C., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural Computation, 13*, 1443–1471.

Schreijäg, S. (2012). *Microstructure and mechanical behavior of deep drawing DC04 steel at different length scales.* PhD thesis, Fakultät für Maschinenbau des Karlsruher Instituts für Technologie (KIT).

Simpson, T. W., Poplinski, J., Koch, P. N., & Allen, J. K. (2001). Metamodels for computer-based engineering design: Survey and recommendations. *Engineering with Computers, 17*(2), 129–150.

Storn, R., & Price, K. (1997). Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization, 11*(4), 341–359.

Tan, R. K., Zhang, N. L., & Ye, W. (2020). A deep learning-based method for the design of microstructural materials. *Structural and Multidisciplinary Optimization, 61*, 1417–1438.

Tax, D. M. J., & Duin, R. P. W. (2004). Support vector data description. *Machine Learning, 54*, 45–66.

Tome, C., Canova, G. R., Kocks, U. F., Christodoulou, N., & Jonas, J. J. (1984). The relation between macroscopic and microscopic strain hardening in f.c.c. polycrystals. *Acta Metallurgica, 32*(10), 1637–1653.

Utkin, L. V., Zaborovsky, V. S., Lukashin, A. A., Popov, S. G., & Podolskaja, A. V. (2017). A Siamese autoencoder preserving distances for anomaly detection in multi-robot systems. In *2017 international conference on control, artificial intelligence, robotics & optimization* (*ICCAIRO*) (pp. 39–44). IEEE.

Van Der Maaten, L., Postma, E., Van den Herik, J., et al. (2009). Dimensionality reduction: A comparative. *Journal of Machine Learning Research, 10*(66–71), 13.

Von Schlippenbach, U., Emren, F., & Lücke, K. (1986). Investigation of the development of the cold rolling texture in deep drawing steels by ODF-analysis. *Acta Metallurgica, 34*(7), 1289–1301.

Zhang, J., & Sanderson, A. C. (2009). Jade: Adaptive differential evolution with optional external archive. *IEEE Transactions on Evolutionary Computation, 13*(5), 945–958.