Modeling deformation behavior of HEAs using contrastive conditional generative adversarial networks

Garima Jain
TCS Research
Tata Consultancy Services Ltd.
Pune, Maharashtra, India
j.garima@tcs.com

Avadhut Sardeshmukh
TCS Research
Tata Consultancy Services Ltd.
Pune, Maharashtra, India
avadhut.sardeshmukh@tcs.com

Gerald Tennyson
TCS Research
Tata Consultancy Services Ltd.
Pune, Maharashtra, India
g.tennyson@tcs.com

Shalini Koneru
TCS Research
Tata Consultancy Services Ltd.
Pune, Maharashtra, India
shaliniroy.koneru@tcs.com

M.R. Rahul
Affiliation
Address
rahulmr@iitism.ac.in

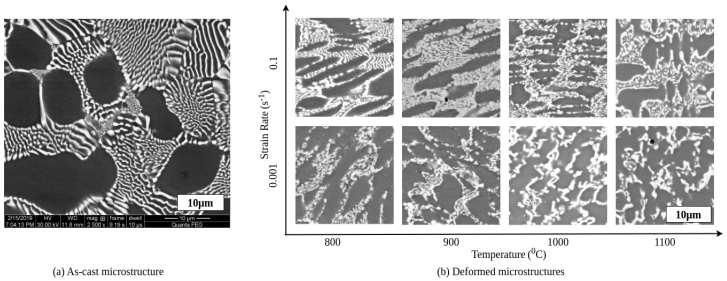
Abstract

High entropy alloys (HEAs) tend to exhibit good mechanical properties, making them potential candidates for various applications. However, tailoring the alloys for target properties requires extensive exploration of microstructure configurations and corresponding properties either through experiments or numerical simulations. Leveraging recent advances in generative modeling, the deformation behavior of $CoCrFeNiTa_{0.395}$ alloy is modeled as conditional generation of deformed microstructures based on processing conditions. To achieve this, a Conditional Generative Adversarial Network (CGAN) model is developed, which synthesizes a deformed microstructure based on temperature and strain rate parameters. A contrastive conditional loss is utilized to induce similarity bias which effectively deals with data sparsity. To help the model learn intricate features across a wide range of process parameters, additional architectural mechanisms like self-attention are employed. Our evaluations reveal good qualitative and quantitative similarities between experimental and predicted microstructures. We also propose a modified contrastive loss for continuous conditioning variables and briefly discuss the ongoing work on demonstrating its generalization capability.

1 Introduction

Traditional alloys such as Ni-based superalloys, Ti-alloys and steels are designed by adding alloy elements to one principal element. However, in the last two decades, a new class of alloys namely High entropy alloys (HEAs), also known as concentrated alloys, which are designed with multiple principal elements have received a lot of attention due to their superior properties. Some examples include FeCoNiCrTa, FeCoCrNiMn, and $Al_{0.5}$MoNbTa. The multiple principal element alloying strategy opens up a vast multi-dimensional compositional and processing space, making it possible to design materials with exceptional mechanical properties and thermal stability. However, extensive experimentation is required to identify favorable thermo-mechanical processing domains for industrial applications, which could be infeasible due to the high time and cost involved. While physics-based multi-scale simulation models are being developed to reduce the need for extensive experimentation, the computational cost of these models is often prohibitively high.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).



(a) As-cast microstructure (b) Deformed microstructures

Figure 1: As-cast and deformed microstructures of $CoCrFeNiTa_{0.395}$ under different strain rates and temperatures

Recent advances in deep generative models can be adapted for faster and reasonably accurate prediction of microstructural changes under given processing conditions, thereby enabling faster screening. Recently, conditional generation models such as CGANs (Conditional Generative Adversarial Networks) have been explored for developing process-structure linkages in various materials [3, 7, 11, 14]. A similar approach can be adapted to identify favorable processing domains for an HEA, by modeling its deformation behavior under given processing conditions. However, the approach comes with several challenges and research gaps. First, well curated experimental data on HEAs is scarce and extremely sparse. For example, consider a recent study [13] on deformation behavior of an HEA $CoCrFeNiTa_{0.395}$. Fig. 1a shows the as-cast microstructure, and fig. 1b shows the deformed microstructures after 50% compression at two strain rates and four temperatures. Given the complex interplay of different phenomena at different processing conditions (e.g., dislocation mobility, microstructural evolution such as coarsening at high temperatures etc.,) the resulting microstructures display a wide variety of features. Training a deep model capable of learning how the processing conditions impact the microstructures is extremely difficult with such limited data. Second, most existing works on conditional generation [4, 5, 7] treat the processing conditions as discrete variables, making it difficult to generalize to new processing conditions which is the main purpose in this application.

In this work, we propose a Contrastive Conditional GAN model for predicting deformed microstructures of $CoCrFeNiTa_{0.395}$ from processing conditions, leveraging an inductive similarity bias based on conditioning labels to reduce the data requirement. We show that the (deformed) microstructures generated by our model are qualitatively and quanitatively similar to the corresponding real ones. We also present an extension of the present model to continuous conditioning variables and briefly discuss the ongoing work on a larger simulated dataset to demonstrate the generalization capability of our model in section 4.

2 Materials and Methods

Shah et al. [13] designed a hypo-eutectic high entropy alloy $CoCrFeNiTa_{0.395}$ with primary dendritic FCC phase and fine eutectic (FCC solution + Laves phase) microstructure possessing good high-temperature mechanical properties. The dark grey phase in the microstructure in fig. 1a is an FCC solid solution with almost equal atomic fractions of Fe, Cr, Ni and Co, while the bright phase in the eutectic is an intermetallic relatively rich in Ta.[1] A high-temperature uniaxial compression test (up to 50% compression) was performed at different temperatures and strain rates(fig. 1b shows the deformed microstructures). The authors observed that deformation at lower temperatures lead to microcracks whereas at high temperature (and lower strain rates) there was a coarsening and loss of lamellar morphology (the black lines inside the white regions in the fig. 1a are the lamellae). At high temperature and high strain rate, the eutectic inter-lamellar spacing increased, but the microstructure is still lamellar in appearance.[2]

Scarcity of training data (only one microstructure available at each processing condition) makes it quite challenging to train a conditional generative model on this data. Vanilla conditional GAN [9]

[1] Here, note that Co and Ni are distributed relatively uniformly across both the phases.

[2] Some of these variations (e.g. inter-lammellar spacing) cannot be clearly obversed at the length scale of the microstructures shown in fig. 1.

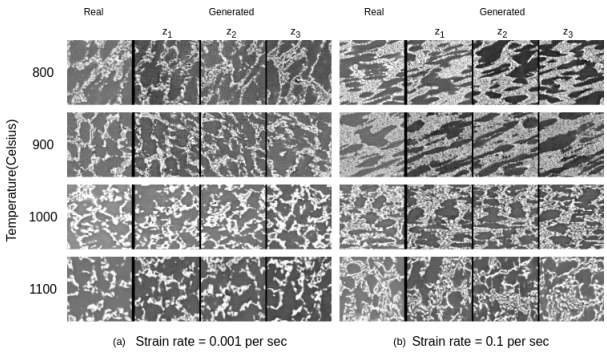


Figure 2: Comparing generated microstructures with real ones for discrete conditional labels.

formulation cannot capture the variations in features at different processing conditions well. Hence, we leverage the contrastive conditional GAN [8] objective function, which is based on the idea of contrastive learning - pull similar inputs close and push disssimilar inputs far apart. By minimizing this objective, the Discriminator learns a representation space in which images with same class label are mapped close together. And the Generator exploits this knowledge to generate images that follow these relationships, reducing the data requirement through inductive bias. Please see supplementary material A.1.1 for the ContraGAN objective function and its intuition.

Further, we replace the Discriminator with a Critic, as in the Wasserstein GAN [1]. By removing the adversarial training, WGAN avoids common GAN training issues such as Discriminator overfitting, mode collapse and lack of diversity. We also add self-attention layer in Generator network(first used in GANs by [16]), which has successfully shown to capture long-range dependencies in image regions. Please see supplementary material A.2 and A.3 for details of the architecture and training.

3 Experimental Results

We trained a ContraGAN model using the loss function in equation 3, treating the strain rate and temperature combinations as discrete class labels. Evaluations are presented in this section.

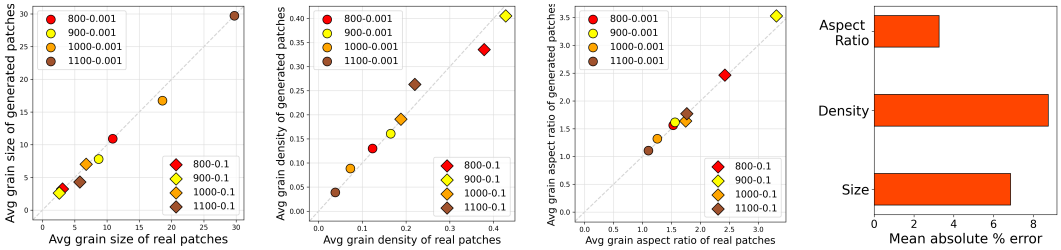
Qualitative Evaluation

Fig. 2 shows for each combination of strain rate and temperature parameters, the real microstructure and three generated microstructures resulting from noise vectors z_1, z_2, z_3 input to the Generator. Note that for the same input noise vector, with different processing conditions, the model is able to generate different microstructures, appropriate for those processing conditions. For example, the second column in fig. 2a shows the different microstructures generated for each tempearture (strain rate 0.001) using the same z_1. Whereas, by combining the same processing condition with different noise vectors (consider any row in fig 2a or b), the model generates variations within the microstructure morphology for that processing condition. Further, all generated microstructures are visually similar to the corresponding real ones.

Quantitative Evaluation

The HEA microstructures contain the primary phase and fine eutectics (which contain FCC and Laves phase), as discussed earlier. The features of interest are the shape and size descriptors, and density of primary phase grains, and features of eutectics such as interlamellar spacing and width of lamella. The finer features such as lamellar width and spacing are unresolvable at the length scale we are working with (a choice driven by data availability). Hence, we present a quantitative comparison between real and generated microstructures based on features of primary phase. The features are computed using a segmentation algorithm (ImageJ [12]) to fit ellipses around each grain.

Figures 3a, 3b, and 3c show the plots of average size, density and aspect ratio, respectively, of real vs generated microstructures for the eight processing conditions. All the points in the three scatter plots lie very close to $x = y$ line, indicating that the average values of features are approximately equal for real and generated microstructures. The average percentage deviations (over all processing conditions), as shown in fig. 3d are well below 10%.



(a) (b) (c) (d)

Figure 3: Average features of primary phase grains in real vs generated microstructures - (a)size, (b)density and (c)aspect ratio. (d) Mean absolute percentage deviation averaged over all processing conditions.

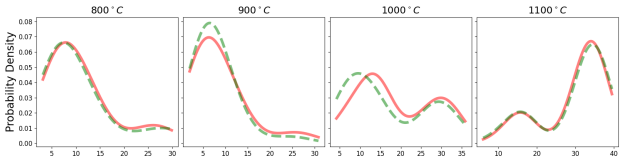


Figure 4: Distributions of average size of primary phase grains in real(read solid line) vs generated(green dashed line) microstructures, for different temperatures at strain rate $0.001s^{-1}$

Further, each part in fig. 4 shows the distributions of grain size in real and generated microstructures for each temperature value at strain rate $0.001s^{-1}$. The distributions match quite well, with the tails also matching. Please see supplementary material A.5 for similar results obtained for aspect ratio at strain rate $0.001s^{-1}$ and both features at strain rate $0.1s^{-1}$.

4 Continuous conditional information

While the current model can predict the microstructures for known cases (which the model *saw* during training) excellently, it can't generalize to new, unseen processing conditions. For this purpose, we modified the contrastive loss function of ContraGAN (3) for continuous conditional variables. Specifically, we propose the following contrastive objective instead of equation 3:

$$\mathcal{L}_{2C} = \frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{m}\mathbf{1}_{j\neq i}\cdot max(0, \delta - ||d(l(x_i), l(x_j)) - \alpha \cdot d(e(y_i), e(y_j))||_2^2) \tag{1}$$

where $\langle x_i, y_i\rangle$ for $i = 1 \ldots m$ is the batch of images and processing conditions, $l(.)$ is the image representation, and $e(.)$ is the process parameter representation learned by the Discriminator, $d(\cdot)$ is a distance measure like cosine or L_2, α is the proportionality constant and δ is a softness parameter. [3] Intuitively, this loss encourages that the distance between image embeddings is proportional to the distance between corresponding processing conditions. The discrete ContraGAN loss only *maximizes* the distance between images with different conditioning labels. In our case, the conditioning variables have numerical significance and we want the distances between image representations to correspond to the distances between conditioning variables.

The current dataset is not a good candidate to evaluate the continuous ContraGAN formulation since it is extremely sparse and doesn't cover the vast continuous space of process parameters. The 2 strain rates $0.001s^{-1}$ and $0.1s^{-1}$ are orders of magnitude apart, with no data points in between. We are currently working on evaluating the modified ContraGAN model on a larger simulated dataset on spinodal decomposition of $Mg_2Si_xSn_{1-x}$ [2].

[3] Note that α and δ are hyperparameters.

5 Summary

In this work, we introduced a conditional generative model to predict deformed microstructures of a high entropy alloy $CoCrFeNiTa_{0.395}$, by treating process parameters as class labels. The generated microstructures were shown to be qualitatively and quantitatively similar to the real ones. We also proposed a continuous form of conditional contrastive loss function, which will be used to train a conditional generative model on continuous conditional information. The model will be trained on a larger, simulated dataset, instead of sparsely available alloy microstructures data. This will show the generalization capability of our model to generate samples for unobserved processing conditions, providing a means for faster identification of safe and favorable processing domains for HEAs.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. "Wasserstein generative adversarial networks". In: *International conference on machine learning*. PMLR. 2017, pp. 214–223.

[2] Vahid Attari, Pejman Honarmandi, Thien Duong, Daniel J Sauceda, Douglas Allaire, and Raymundo Arroyave. "Uncertainty propagation in a multiscale CALPHAD-reinforced elastochemical phase-field model". In: *Acta Materialia* 183 (2020), pp. 452–470.

[3] Lars Banko, Yury Lysogorskiy, Dario Grochla, Dennis Naujoks, Ralf Drautz, and Alfred Ludwig. "Predicting structure zone diagrams for thin film synthesis by generative machine learning". In: *Communications Materials* 1.1 (Mar. 2020), p. 15. ISSN: 2662-4443. DOI: 10.1038/s43246-020-0017-2. URL: https://doi.org/10.1038/s43246-020-0017-2.

[4] Lars Banko, Yury Lysogorskiy, Dario Grochla, Dennis Naujoks, Ralf Drautz, and Alfred Ludwig. "Predicting structure zone diagrams for thin film synthesis by generative machine learning". In: *Communications Materials* 1.1 (2020), p. 15.

[5] Arthur Baucour, Myungjoon Kim, and Jonghwa Shin. "Data-driven concurrent nanostructure optimization based on conditional generative adversarial networks". In: *Nanophotonics* 11.12 (2022), pp. 2865–2873.

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial networks". In: *Communications of the ACM* 63.11 (2020), pp. 139–144.

[7] Akshay Iyer, Biswadip Dey, Arindam Dasgupta, Wei Chen, and Amit Chakraborty. "A conditional generative model for predicting material microstructures from processing methods". In: *arXiv preprint arXiv:1910.02133* (2019).

[8] Minguk Kang and Jaesik Park. "ContraGAN: Contrastive Learning for Conditional Image Generation". In: *arXiv e-prints*, arXiv:2006.12681 (June 2020), arXiv:2006.12681. DOI: 10.48550/arXiv.2006.12681. arXiv: 2006.12681 [cs.CV].

[9] Mehdi Mirza and Simon Osindero. "Conditional Generative Adversarial Nets". In: *CoRR* abs/1411.1784 (2014). arXiv: 1411.1784. URL: http://arxiv.org/abs/1411.1784.

[10] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Curran Associates, Inc., 2019, pp. 8024–8035.

[11] Avadhut Sardeshmukh, Garima Jain, Sreedhar Reddy, B. P. Gautham, K. V. Vamsi, Pushpak Bhattacharyya, and Ujjal Tewary. "Development of Process-Structure Linkage Using Conditional Generative Adversarial Networks". In: *Metallurgical and Materials Transactions A* 55.7 (July 2024), pp. 2213–2229. DOI: 10.1007/s11661-024-07386-9.

[12] Caroline A Schneider, Wayne S Rasband, and Kevin W Eliceiri. "NIH Image to ImageJ: 25 years of image analysis". In: *Nature methods* 9.7 (2012), pp. 671–675.

[13] Naishalkumar Shah, MR Rahul, Sandip Bysakh, and Gandham Phanikumar. "Microstructure stability during high temperature deformation of CoCrFeNiTa eutectic high entropy alloy through nano-scale precipitation". In: *Materials Science and Engineering: A* 824 (2021), p. 141793.

[14] Jianan Tang, Xiao Geng, Dongsheng Li, Yunfeng Shi, Jianhua Tong, Hai Xiao, and Fei Peng. "Machine learning-based microstructure prediction during laser sintering of alumina". In: *Scientific Reports* 11.1 (2021), pp. 1–10.

[15] A Vaswani. "Attention is all you need". In: *Advances in Neural Information Processing Systems* (2017).

[16] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. "Self-attention generative adversarial networks". In: *International conference on machine learning*. PMLR. 2019, pp. 7354–7363.

A Supplementary material

A.1 Conditional Wasserstein GAN

Generative Adversarial Network(GAN) [6] belongs to a class of generative models, which is used to generate synthetic data by learning distribution of their training data. It consists of two competing networks - Generator and Discriminator, which are trained using a minimax game. Generator transforms an input noise vector(from a Gaussian or Uniform distribution) to a sample from target data distribution, trying to mimic real data distribution as closely as possible. Discriminator takes in real and generated data samples, and computes Jenson-Shannon divergence between real and generated distributions. Generator is trained to fool the Discriminator into thinking that it generated a real data sample, while the Discriminator is trained to distinguish the generated samples from the real samples. Because of this adversarial training, GANs are difficult to train.

To overcome many problems posed by vanilla GAN formulation(like mode collapse and vanishing gradients), Wasserstein GAN was introduced [1], which computes Wasserstein's distance between the two distributions. Equation 2 gives the objective function, which the Critic(since Discriminator is no longer an adversary, it's called Critic) tries to maximize, while the Generator tries to minimize it:

$$\mathbb{E}_{x\sim P_r}[f_w(x)] - \mathbb{E}_{z\sim p(z)}[f_w(g_\theta(z))] \tag{2}$$

where, x is a sample from real data distribution P_r, z is a noise vector from standard Gaussian distribution $\mathcal{N}(0, \mathbf{I})$, f_w represents the Critic function, and g_θ is the Generator.

The Generators trained using above GAN formulations are limited to generating random data from the learned data distributions, and cannot sample from specific regions of the distribution. To generate a particular kind of samples(for example, to generate images of digit 5 by training on MNIST dataset), a conditional variant of GAN was introduced by [9], which allows to conditionally generate samples. Feeding conditional information to the Generator, along with a noise vector, allows the model to generate samples appropriate to the fed conditional information.

We used a conditional Wasserstein GAN formulation to train conditional generative model, by giving class labels as conditional information.

A.1.1 Conditional contrastive loss

A contrastive learning method for conditional image distribution was introduced by [8], which presented a conditional contrastive loss:

$$\mathcal{L}_{2C} = \frac{1}{m}\sum_{i=1}^{m} -log\left(\frac{e^{l(x_i)^T e(y_i)/t} + \sum_{k=1}^{m}\mathbf{1}_{y_k=y_i}\cdot e^{l(x_i)^T l(x_k)/t}}{e^{l(x_i)^T e(y_i)/t} + \sum_{k=1}^{m}\mathbf{1}_{k\neq i}\cdot e^{l(x_i)^T l(x_k)/t}}\right) \tag{3}$$

where, $\langle x_i, y_i\rangle$ for $i = 1 \ldots m$ is the batch of images and conditioning labels, $l(x_i)$ is the image representation and $e(y_i)$ is the class (discrete conditioning variable) representation learned by the Discriminator.

Intuitively, in a batch, the loss function pulls the image representations together if their conditioning labels are same(similar samples) and pushes them apart otherwise(dissimilar samples). It also brings the image representations and the corresponding class representations close.

The loss function given by equation 3 was added to the usual Wasserstein GAN objective(equation 2), to train Critic and Generator networks.

A.2 Model architecture

Figure S-1 shows high-level architecture of our proposed Conditional Generative adversarial network model. The model consists of two networks: Generator and Critic - Generator takes in a Gaussian noise vector, $z \sim \mathcal{N}(0, \mathbf{I})$ and conditional information (the processing conditions in our case) and generates microstructures appropriate for those processing conditions; while Critic takes either a batch of real microstructures or generated microstructures, combined with corresponding conditional information (processing conditions), and outputs an approximation of the Wasserstein's distance between the data (real) and Generator's distribution. The Critic network also outputs the image and processing condition (treated as a class label) representations, which are used to compute the contrastive loss.

We explored different ways of feeding processing conditions (treated as class labels) to the Generator and Critic. Both Generator and Critic contain an embedding layer, which maps the integer class label to a sparse embedding (one-hot vector), with dimension equal to the number of classes (in our case, total number of processing conditions, i.e. 8) in the dataset. The Generator then linearly projects the embedding to a vector,

whose dimension is same as that of Gaussian noise vector, which is then multiplied element-wise with the noise, and passed to the next layers. The Critic linearly projects the embedding to a high-dimensional space (equal to the feature size of fourth convolution layer), which is then reshaped and appended to the output of fourth convolution layer.

In Generator, we also used self-attention layer (64 features) [15] after third convolution transpose layer, which helps the network learn and generate intricate features of the microstructures. Tables1 and 2 give input and output feature sizes of each layer in Generator and Critic network respectively. Tangent hyperbolic(TanH) activation function was used at Generator's output layer, which scales output in the range $[-1, 1]$, and no activation function was used at Critic's output layer.

In both networks, kernel size of each convolution layer is 4×4, with stride 2 and padding 1, and each batch norm layer is followed by Leaky Rectified Linear Unit(ReLU) activation function($slope = 0.2$).

Layer	Input feature size	Output feature size
Embedding	8	128
Linear	128	8192
Convolution Transpose I	512 x (4 x 4)	256 x (8 x 8)
Batch Norm	256 x (8 x 8)	256 x (8 x 8)
Convolution Transpose II	256 x (8 x 8)	128 x (16 x 16)
Batch Norm	128 x (16 x 16)	128 x (16 x 16)
Convolution Transpose III	128 x (16 x 16)	64 x (32 x 32)
Batch Norm	64 x (32 x 32)	64 x (32 x 32)
Self-attention	64 x (32 x 32)	64 x (32 x 32)
Convolution Transpose IV	64 x (32 x 32)	64 x (64 x 64)
Batch Norm	64 x (64 x 64)	64 x (64 x 64)
Convolution Transpose V	64 x (64 x 64)	1 x (128 x 128)
Batch Norm(Output)	1 x (128 x 128)	1 x (128 x 128)

Table 1: Details of each layer in Generator network

Layer	Input feature size	Output featue size
Embedding	8	64
Convolution I	1 x (128 x 128)	64 x (64 x 64)
Convolution II	64 x (64 x 64)	64 x (32 x 32)
Batch Norm	64 x (32 x 32)	64 x (32 x 32)
Convolution III	64 x (32 x 32)	128 x (16 x 16)
Batch Norm	128 x (16 x 16)	128 x (16 x 16)
Convolution IV	128 x (16 x 16)	256 x (8 x 8)
Batch Norm	256 x (8 x 8)	256 x (8 x 8)
Convolution V	257 x (8 x 8)	128 x (4 x 4)
Batch Norm	128 x (4 x 4)	128 x (4 x 4)
Linear(Output)	2048	1
Projection Layer I	2048	1024
Projection Layer II(Output)	1024	64

Table 2: Details of each layer in Critic network

A.3 Training details

Our CGAN model was implemented and trained using PyTorch [10] library. For optimization, we employed the Root Mean Squared Propagation(RMSProp) algorithm with a mini-batch size of 128. The model was trained for 800 epochs.

We used Bayesian optimization algorithm to find the optimal values of following hyperparameters: learning rate of the RMSProp optimizer, weight clipping range used to constrain Critic's weights to enforce Lipschitz continuity, number of times Critic is updated as compared to Generator, and the slope of Leaky ReLU activation function. Table3 gives values of these and other hyperparameters. Weight of contrastive loss function and the temperature constant were tuned manually.

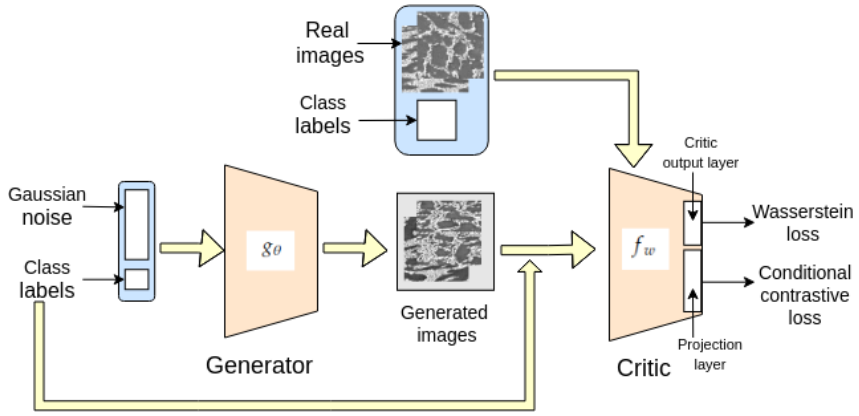


Figure S-1: High level architecture of our proposed Contrastive conditional GAN model

Hyperparameter	Value
Learning rate of optimizer	0.0002
Critic weight clipping range	[-0.009, 0.009]
No of times Critic is updated	4
Leaky ReLU slope	0.2
Contrastive loss function weight	1
Contrastive loss function temperature	0.1

Table 3: Values of hyperparameters

A.4 Data pre-processing

As a first pre-processing step, microstructures containing cracks were discarded, like shown in fig. S-2. Also, from some higher resolution microstructures showing cracks, regions of interest were manually cropped, avoiding the area around cracks, since the deformation in those areas could have been governed by different phenomena. Lastly, we used image processing techniques to bring all the microstructures at a uniform contrast level. The original microstructures were captured at different magnifications. To get a sufficiently large training dataset, we cropped patches from the original microstructures, such that all patches are at a uniform length scale (they represent an area of $100\mu m^2$) and the relevant microstructural features are clearly visible. After cropping patches, we obtained around 1200 images, and applied some techniques to augment the dataset, to get our training dataset. Finally, all images in the dataset were scaled to $[-1, 1]$ range.

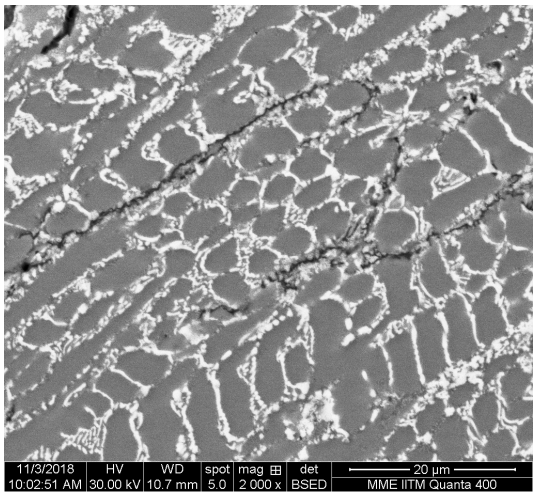
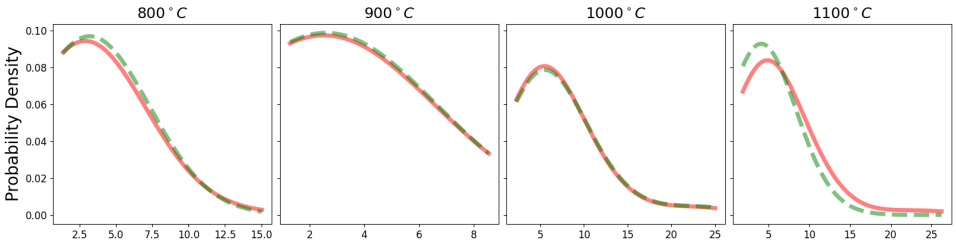
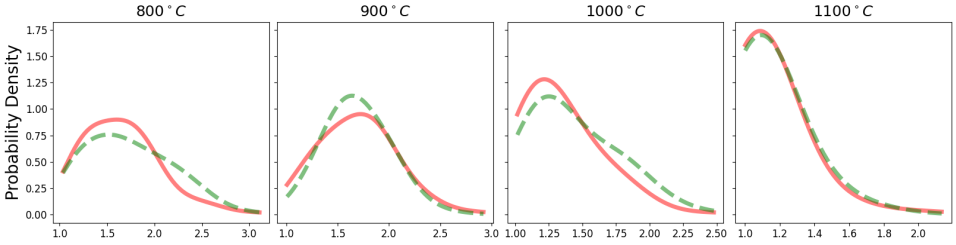


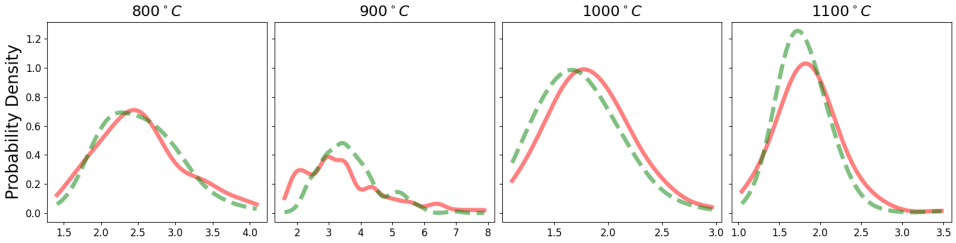
Figure S-2: An example microstructure containing crack.



(a) Distributions of average size of primary phase grains for every temperature at strain rate $0.1s^{-1}$



(b) Distributions of aspect ratio of primary phase grains for every temperature at strain rate $0.001s^{-1}$



(c) Distributions of aspect ratio of primary phase grains for every temperature at strain rate $0.1s^{-1}$

Figure S-3: Distributions of features of primary phase grains in real (read solid line) vs generated (green dashed line) microstructures

A.5 Additional Experimental Results

In the main paper(fig. 4) we showed that the average size distributions of generated microstructures match with corresponding real ones, for strain rate 0.001. Figure S-3 below shows similar results for average size of strain rate 0.001, and average size and aspect ratio features for strain rate 0.1. The figures show that feature distributions match very well, with an exception in aspect ratio distribution for $900^{\circ}C, 0.1s^{-1}$(i.e. second plot in fig. S-3c). Perhaps this is due to the large variation in the aspect ratio of primary phase grains for this processing condition. Figure S-4 shows the real deformed microstructure in that case. While most of the grains are elongated (due to the compression), there are many smaller grains which are not affected/deformed, resulting in multiple peaks in the distribution. The generated distribution ends up averaging some of the nearby peaks which leads to a mismatch between the two distributions.

A.5.1 Ablation Study

We performed an ablation to study how conditional contrastive loss helps to deal with data scarcity problem effectively. We trained a generative model, with just Wasserstein GAN loss(eq. 2), and removed conditional contrastive loss(eq. 3), keeping all other settings same. Figure S-5 shows the microstructures generated by this model, compared with corresponding real microstructures at each processing condition. Qualitatively, the generated microstructures look good, but they do not necessarily correspond to the appropriate processing condition, indicating that the model is not able to distinguish among different classes(processing conditions) very well. The model is learning the overall data distribution, but is unable to learn the conditional distribution of microstructures, region where we would like to sample from.

We believe this is a consequence of working in a low data regime(our dataset consists of just 1 microstructure at each processing condition). In the absence of sufficient data, the model is not able to fully capture the distinguishing microstructural features at each processing condition. The class labels which are input to the Discriminator and Generator models during training, could have provided enough implicit signal to differentiate between classes provided there was enough data. The conditional contrastive loss explicitly tries to pull together

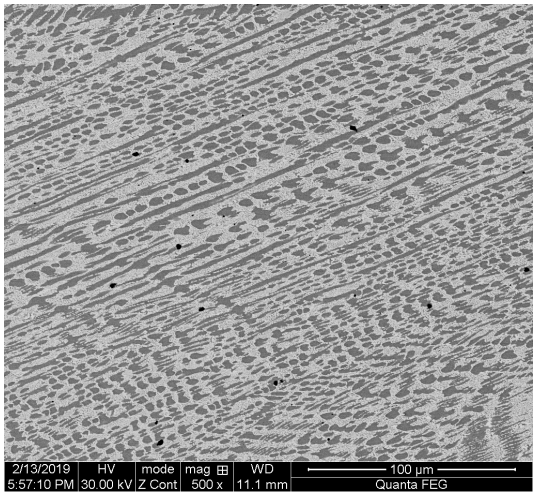


Figure S-4: A microstructure from processing condition $900°C, 0.1s^{-1}$, showing variations in aspect ratio of primary phase grains.

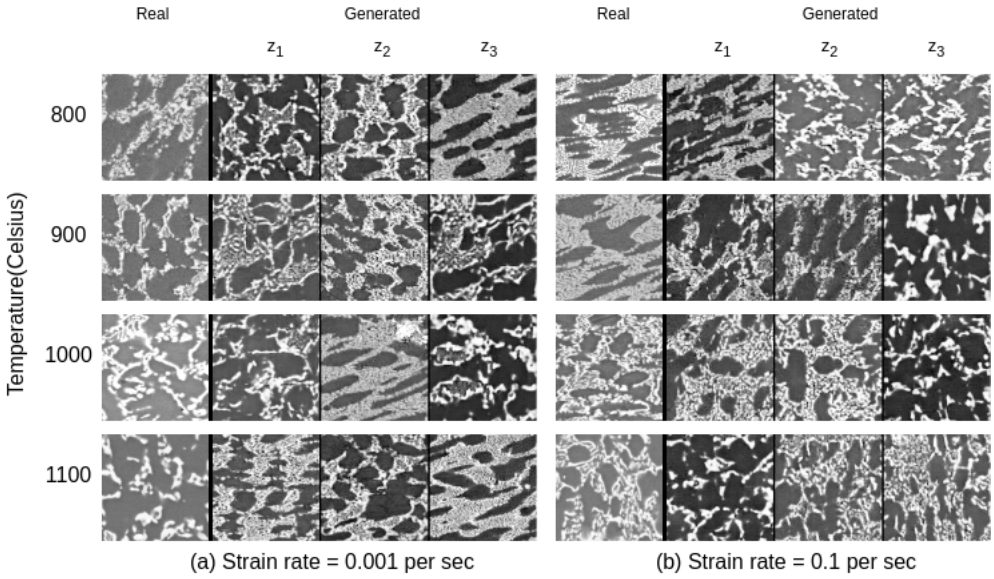


Figure S-5: Ablation study: Comparing generated microstructures with real ones without using conditional contrastive loss

samples belonging to same class, and pushes apart those belonging to different classes in the representation space, hence learning the class boundaries well. This sort of provides an inductive bias based on similarity measure in the class and image representation space.