ARL-TR-9734 • AUG 2023

A Generative Adversarial Network Approach with a Random Patch Discriminator to Generate 3D Synthetic Microstructures Containing Second-Phase Particles

by P Justin McKee and Jeffrey T Lloyd

NOTICES

Disclaimers

ARL-TR-9734 ● AUG 2023

A Generative Adversarial Network Approach with a Random Patch Discriminator to Generate 3D Synthetic Microstructures Containing Second-Phase Particles

P Justin McKee and Jeffrey T Lloyd
DEVCOM Army Research Laboratory

REPORT DOCUMENTATION PAGE

1. REPORT DATE	**2. REPORT TYPE**	**3. DATES COVERED**	
		START DATE	**END DATE**
August 2023	Technical Report	October 1, 2020	September 30, 2021

4. TITLE AND SUBTITLE
A Generative Adversarial Network Approach with a Random Patch Discriminator to Generate 3D Synthetic Microstructures Containing Second-Phase Particles

5a. CONTRACT NUMBER	**5b. GRANT NUMBER**	**5c. PROGRAM ELEMENT NUMBER**
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**

6. AUTHOR(S)
P Justin McKee and Jeffrey T Lloyd

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	**8. PERFORMING ORGANIZATION REPORT NUMBER**
DEVCOM Army Research Laboratory ATTN: FCDD-RLA-TB Aberdeen Proving Ground, MD 21005	ARL-TR-9734

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	**10. SPONSOR/MONITOR'S ACRONYM(S)**	**11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

13. SUPPLEMENTARY NOTES
ORCID ID(s): Jeffrey Lloyd, 0000-0002-4930-5361

14. ABSTRACT

The failure of metals under quasi-static and dynamic loads is influenced by second-phase particles that act as failure nucleation sites. It is desirable to have a large quantity of data when developing models to understand how particles contribute to failure, but collection of such data from real samples is prohibitively costly and time consuming. In this work, a machine learning approach is developed to learn the features of a microcomputed tomography dataset and output 3D synthetic particle microstructure volumes. The machine learning approach comprises a generative adversarial network in conjunction with the random patch discriminator method. The method maintains training stability when generating volumes up to 128×128×128 voxels by using a random patch method for the discriminator model. This random patch method takes advantage of the ability to sample smaller sections of the whole data to determine whether characteristics within the samples are representative of the microstructure. Generated volumes are compared against the real dataset to demonstrate the ability to capture all features of the sparse particles when trained from a relatively small set of data. Statistical accuracy is verified by comparing the distribution of volume, size, and shape of the particles, as well as two-point and clustering correlation functions. We show that our machine learning approach is particularly well suited for describing two-phase data with disparate phase fractions and data that possess anisotropy in the size, shape, and spacing of individual phases compared to other approaches that describe two-phase materials with similar volume fractions and more isotropic features.

15. SUBJECT TERMS
Sciences of Extreme Materials, machine learning, synthetic microstructure generation, generative adversarial network, second-phase particles

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**
a. REPORT	**b. ABSTRACT**	**C. THIS PAGE**		
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UU	32

19a. NAME OF RESPONSIBLE PERSON	**19b. PHONE NUMBER** *(Include area code)*
Philip McKee	(410) 278-3031

STANDARD FORM 298 (REV. 5/2020)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

List of Tables

Acknowledgments

We thank Professor Andrew Stuart and Mr Nikola Kovachki (Caltech) for helpful discussions about different approaches regarding implementation of the generative adversarial network. We also thank Dr Saket Thapliyal (University of North Texas) for acquiring the raw microstructure X-ray computed tomography data.

1. Introduction

Failure and the response of aluminum during plastic deformation is influenced by the presence of second-phase particles that are distributed through the matrix. Ductile failure occurs with cracking of larger particles, nucleation of voids, and coalescence of void sheets.[1–4] Ballistic impacts show that the alignment of particles in aluminum can alter the failure mechanism, where ledges in the fracture surface appear to follow the lines of particles.[5] However, there is still limited knowledge on how particle distribution statistics relate to the performance of aluminum alloys. Models have been developed to simulate softening that represents void growth, but these models use an idealistic distribution and shape of particles.[6–8] Development of homogenized models that can represent the statistical response of a realistic particle distribution require a large amount of data for any combination of material and processing history. Collecting and processing X-ray computed tomography (CT) scan data directly for each sample is prohibitively time consuming and costly. Therefore, we have developed a machine learning-based approach to generate a synthetic microstructure volume that is statistically comparable to a set of training data. These generated volumes provide (1) the microstructure data needed to assist with material model development and (2) a better understanding of how the microstructure contributes to failure under a variety of loading scenarios.

Although there are methods to generate synthetic, two-phase volumes and images to match reference data, generating second-phase particle volumes present unique challenges. The particles are sparsely distributed within the volume and possess anisotropy in particle shape and clustering patterns within the 3D volume. There is a need to capture both small-scale features of individual particles as well as larger scale positioning and clustering of particles. In addition, there is a need to generate statistically representative microstructures with a limited amount of reference data due to the time and expense associated with collecting CT data, especially when anisotropy prevents data augmentation techniques such as mirroring or rotation. In this work, a machine learning generative adversarial network (GAN) approach is used to generate 3D volumes. However, a conventional GAN method is not able to effectively capture the details of the fine, sparse particles when generating a larger volume. To overcome this limitation, we developed a random patch discriminator method that samples smaller sections within the larger generated or training volume. The random patch discriminator can effectively evaluate a large volume from a limited selection of data because local patterns of the second-phase particle microstructure are still representative of the global volume. In addition, the discriminator can better characterize the presence of small particles when evaluating a reduced sample volume. It can also reduce complications from mode collapse to generate a wide variety of feasible output data even when a relatively small training data set is provided.

In previous work, other researchers used several approaches to generate 2D and 3D representations of microstructures, porous media, or other two-phase structures. One conventional approach is the Yeong–Torquato[9,10] method, which uses simulated annealing. This method is an iterative approach that makes use of correlation functions to modify a starting microstructure until an objective function is satisfied. This approach has demonstrated good results for generating a variety of two-phase structures,[11–14] but it also has limitations. The computational cost to calculate correlation functions on a 3D volume, and particularly for anisotropic data, is high. Lower order correlation functions such as the two-point correlation function, which is often used with this method, contain some information about anisotropy but cannot fully describe a 3D structure. It is possible to reach a result that satisfies the objective function, but it is still a poor representation of the target data. Moreover, the full process must be repeated for each microstructure to be generated, which adds more time to generate a large set of data.

Another approach is the use of a Markov random field, which was first developed for texture synthesis.[15–18] In this method, a probability density function is generated under the assumption that any local window should be able to satisfy the same set of structural descriptors. This method has been applied to material microstructure data to generate both 2D and 3D volumes.[19–22] Although this method improves computational time compared with the Yeong–Torquato method, there is still a requirement to run the process for each microstructure that is generated. The need for an increased sampling window for data with large feature sizes combined with more complex descriptors can still result in significant computational expense per generated sample.

Machine learning is an alternative that offers several methods for microstructure reconstruction, which have become feasible with increasing computer power to train models. In an early application of machine learning, a classification tree model was used to consider local volume and then predict the state of the next missing voxel until a full microstructure was built.[23] This model was effective for generating structures that could be reasonably described by the local volume; however, longer range correlations were more difficult to capture.

A GAN is a generative machine learning approach that has been used for image generation[24] and has also previously been applied to two-phase microstructures in both 2D[25–27] and 3D.[28–31] Two-dimensional images of a microstructure can be successfully generated by using GAN techniques that were developed in other fields. However, 3D volumes present a few unique challenges. A primary concern is stability issues in which the generator will fail to converge or experience mode collapse when attempting to increase the size of the generated volume and the number of trainable parameters required to describe the large volume. The size of the network quickly becomes a problem when dealing with 3D volumes compared with 2D images. Zhang et al.[31] approached this stability problem by using a variational autoencoder to first convert a 2D section of the target microstructure into the input vector for their generator model. They demonstrated that by starting from this meaningful input vector rather than from a random vector, they were able to generate larger volumes.

In this report, we describe a new method to train a GAN to generate larger 3D microstructure volumes and focus on sparse anisotropic microstructure data. Here, we maintain the training stability by using a random patch approach in the discriminator that evaluates reduced portions of the generated and training data to capture the shape and distribution of the smaller particles contained in the volume. The discriminator is effective when evaluating whether a sample is realistic because any randomly sampled section should be representative of the training data due to the nature of the microstructure pattern. By reducing the training parameters in the discriminator, we can generate larger 3D microstructure volumes that represent the training data while simultaneously improving stability and reducing memory usage.

2. Methods

2.1 Network Structure

The GANs, originally proposed by Goodfellow et al.,[24] have commonly been used for 2D image generation, which has rapidly advanced in recent years.[32–34] Convolutional networks are used to efficiently identify and reproduce features and patterns at multiple length scales in pixelated data. The key feature of a GAN that sets it apart from a simple convolutional neural network is the use of both a generator and discriminator model. The generator is the model that will generate a synthetic volume, whereas the discriminator model is used to evaluate whether a given volume belongs to the training dataset. The generator is optimized to create volumes that are determined by the discriminator to be part of the training set. Both models are trained simultaneously to allow continuous improvement, ensuring that the progress of one model does not outpace the other. This approach is notable compared with other generative machine learning models because the discriminator drives the generator to create sharper images or volumes with greater clarity.[34] After the initial training, the generator can then quickly produce a large quantity of volumes with minimal computation time. The structure of the GAN used in this work is based on the deep convolution GAN (DCGAN) method, originally described by Radford et al.,[34] which uses strided convolutions, batch normalization, and an adversarial loss function. Figure 1 shows the structure of the GAN used to generate particle microstructures followed by details of each of its components.

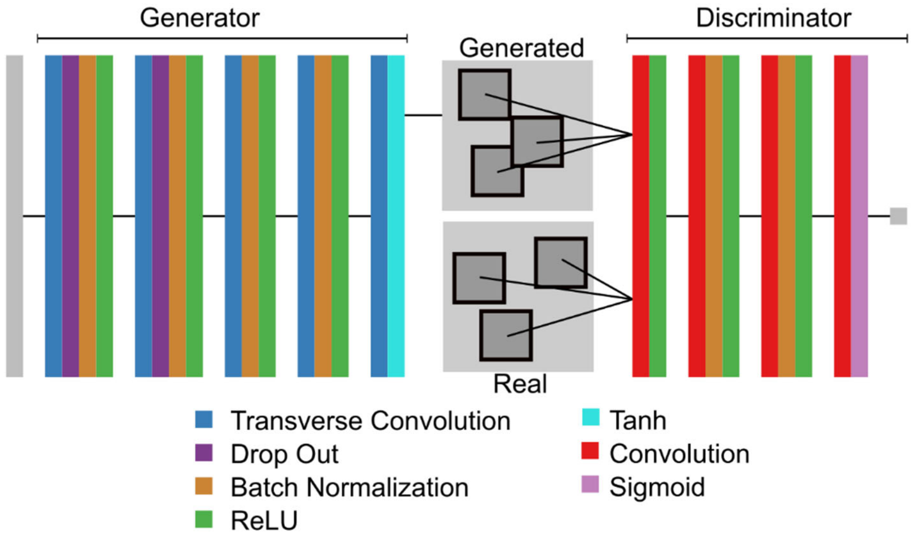

Fig. 1 **Summary of the overall network structure used to generate microstructure volumes. A random vector is converted to a 3D microstructure by the generator. Random patches from generated or real microstructures are evaluated by the discriminator.**

2.2 Generator

The generator network consists of six layers and is based on the generator described by Radford et al.[34] Table 1 shows a summary of network parameters, which produce a 128 × 128 × 128 voxel volume. The initial input to the generator is a vector of random values, which is the source of the variability in the output volume. Layers of the network contain a 3D transverse convolution, batch normalization, and rectified linear unit (ReLU), with the exception of the last layer, which uses a hyperbolic tangent activation function to output values for the generated volume. The first two layers also contain a dropout function set to a value of 0.5 so that 50% of nodes are randomly disabled during training. These dropout layers resulted in a slight improvement of output statistics, but a detailed optimization of the dropout value has not been performed. The initial weight for the convolution and batch normalization parameters are initialized with random Gaussian noise with an average of 0 and 1, respectively, and a standard deviation of 0.2 for both operations.

Table 1 Summary of layers in the generator model

Size in	Size out	Filters in	Filters out	Stride	Kernel size	Padding size	Padding mode	Batch norm	Drop out	Activation
1	4	100	1024	1	4	0	NA	Yes	0.5	ReLU
4	8	1024	512	2	4	1	Zeros	Yes	0.5	ReLU
8	16	512	256	2	4	1	Zeros	Yes	0.0	ReLU
16	32	256	128	2	4	1	Zeros	Yes	0.0	ReLU
32	64	128	64	2	4	1	Zeros	Yes	0.0	ReLU
64	128	64	1	2	4	1	Zeros	No	0.0	Tanh

2.3 Discriminator

The overall architecture of the discriminator is similar to previous GAN networks,[34] consisting of a series of convolutions, batch normalization, and leaky ReLU, with a slope of 0.2, followed by a final convolution and sigmoid activation. Layer details are shown in Table 2. Similar to the generator, the convolution and batch normalization parameters are initialized with random Gaussian noise with an average of 0 and 1, respectively, and a standard deviation of 0.2 for both operations.

Table 2 Summary of layers in the discriminator model

Size in	Size out	Filters in	Filters out	Stride	Kernel size	Padding size	Padding mode	Batch norm	Drop out	Activation
32	4	1	128	2	4	1	Replicate	No	0.0	Leaky ReLU 0.2
16	8	128	256	2	4	1	Replicate	Yes	0.0	Leaky ReLU 0.2
8	16	256	512	2	4	1	Replicate	Yes	0.0	Leaky ReLU 0.2
4	1	512	1	1	4	0	NA	No	0.0	Sigmoid

In contrast to many previous GAN networks, the size of volumes input into the discriminator is smaller than the size of volumes created by the generator. If an attempt is made to pass a full-size volume to be evaluated by the discriminator, this will result in training instability as the size of the training and generated volumes increases. Our attempt to use a patch discriminator method using a grid of patches rather than the full image produced better particle shapes, but the distribution of particles was influenced by the grid spacing. To overcome this issue, we used a random patch approach in which a set of patches taken from random locations within the generated or training volume were passed to the discriminator. The random patch approach helps resolve instability, decouples the size of the discriminator input from the size of the generator output, and reduces memory usage.

A set of five-volume patches of size 32 × 32 × 32 voxels was randomly sampled from generated or training volumes and passed to the discriminator to be processed using 3D convolutions. In addition, we explored the size of the patch and number of patches to better understand how these parameters contribute to the output. Our motivation behind this random patch method is to evaluate the validity of the microstructure using a smaller sample when the features of the microstructure are representative of the larger sample. If any of the random locations can be evaluated and determined to be acceptable, then the whole volume should be acceptable because the particle distribution pattern should be consistent throughout. This sampling assumption is similar to the Markov random field approach for microstructure and texture generation. However, the efficiency of the convolutional network allows for relatively large sampling volumes to be computed efficiently.

2.4 Implementation

The GAN is implemented in PyTorch (version 1.3.1).[35] Training is performed over 300 epochs with 5,000 training loops per epoch and a batch size of 80 samples. Each sample in the batch is taken from a random location in the full training data volume. The Adam optimizer[36] is used with a learning rate of 2.0×10^{-4} and momentum constants β1 and β2 set to 0.5 and 0.999, respectively, to optimize parameters of the network. Both generator and discriminator networks use the same learning rate and are updated at even intervals. Training is performed with six Nvidia V100 32GB GPUs. Training loss is calculated from the adversarial loss given by Goodfellow et al.[24] using the objective function defined as

$$O(X,Z) = E_X(\log D(X)) + E_Z(\log(1 - D(G(Z)))), \qquad (1)$$

where E_X is the label for real data and is set to a value of 1, E_Z is the label for generated data and is set to a value of 0, $D()$ is the discriminator that has an output value between 0 and 1, $G()$ is the generator, X is a volume from the real dataset, and Z is a vector of random values. The models are considered optimized when the generator maximizes the output of the objective function while the discriminator minimizes the output.

In practice, however, it is more computationally efficient to instead calculate a loss function separately for the generator and discriminator that are both minimized. The discriminator loss, L_D, is defined as

$$L_D = [E_X(\log D(X)) + (1 - E_X)\log(1 - D(X))] +$$
$$[E_Z(\log D(G(Z))) + (1 - E_Z)\log(1 - D(G(Z)))], \qquad (2)$$

where the binary cross-entropy is calculated first for the discriminator output from training data compared against the label E_X (first square bracket) and summed with the binary cross-entropy calculated for the discriminator output from generated data compared against the label E_z (second square bracket). The goal of this loss function is to train the discriminator to correctly classify data from the training set as real (label E_X) and data created by the generator as fake (label E_z).

The generator loss, L_G, is calculated at the same time as L_D, and it is defined as

$$L_G = E_X(\log D(G(Z))) + (1 - E_X)\log(1 - D(G(Z))), \tag{3}$$

where the binary cross-entropy is calculated for discriminator output from generated data compared against the label E_X. Here, the goal of optimizing the generator is to create data that the discriminator will classify as real. In contrast to Eq. 2, discriminator output with generated data is now compared against the label for real data (E_X) to drive the generator towards this goal. Loss is calculated for each volume patch and averaged for the overall loss of the sample.

2.5 Dataset

We imaged a 1.25-inch-thick 7075 aluminum alloy plate that had been processed in the T651 condition. The through-thickness variations in second-phase particle distribution arise due to differences in the thermomechanical processing history in regions near the surface, compared to regions along the plate centerline. Material near the surface has more deformation and a faster cooling rate compared to material in the center. The GAN model is trained for two datasets made up of CT image stacks at 1.31-μm resolution, where image contrast is used to identify second-phase particles. The CT data are prepared for use in the GAN by first applying a threshold to segment the particles from the background matrix and binarize the images. Images in the stack are then down sampled by a factor of one third using linear interpolation before training. The volume is not converted back to a binary after the interpolation because we found that binary data caused training to be less stable. The data are normalized between −1 and 1 to allow for more efficient training.

The first dataset is from the center section of the scanned plate and contains fewer, larger particles (shown in Fig. 2). The second dataset is taken closer to the surface of the material where particles are smaller and more numerous (shown in Fig. 3). These datasets will be referred to as the center and surface dataset, respectively. Anisotropy can be seen in the microstructure in both figures with clear differences in both particle shape and spatial distribution between the two datasets. In addition, there are differences in the shape and distribution of particles when viewed from the normal, transverse, and rolling directions. Particles tend to be arranged in lines along the rolling direction. The 2D methods of generation are insufficient to capture the particle details, which can be seen in these figures, as a result of this anisotropy.

The goal is to demonstrate the effectiveness of the microstructure generation process on the two datasets, which each have unique features.

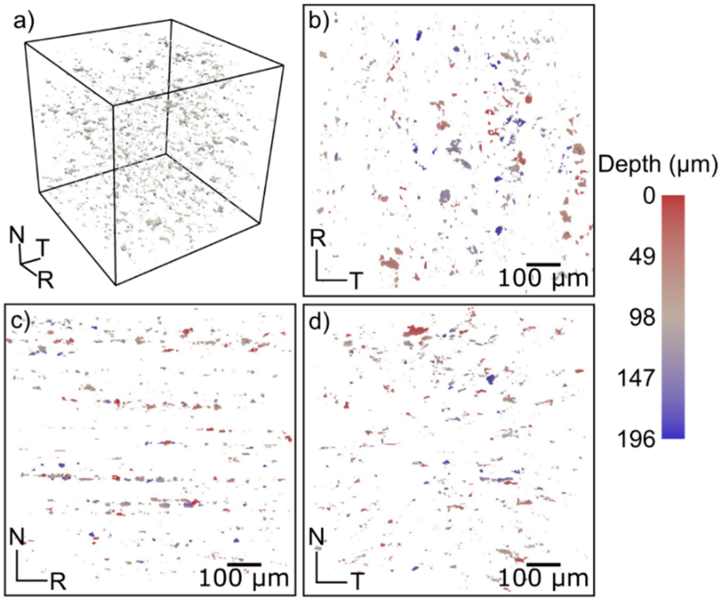

Fig. 2 A portion of the center dataset particle morphology shown by a) 3D view of a 200 × 200 × 200 voxel cube (side length 786 µm). Sections of 200 × 200 × 50 voxels (196 µm) are shown viewed from the b) normal, c) transverse, and d) rolling directions with particles color coded according to their depth through the thickness.

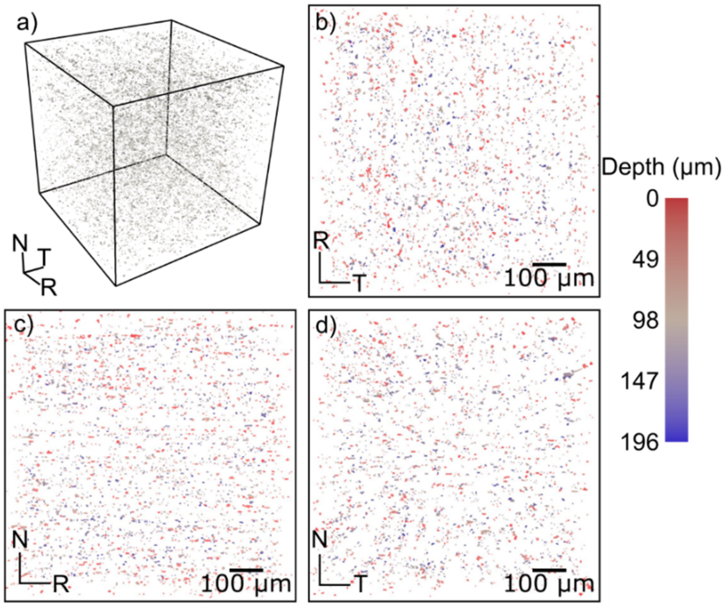

Fig. 3 A portion of the surface dataset particle morphology shown by a) 3D view of a 200 × 200 × 200 voxel cube (side length 786 µm). Sections of 200 × 200 × 50 voxels (196 µm) are shown viewed from the b) normal, c) transverse, and d) rolling directions with particles color coded according to their depth through the thickness.

These volumes present a challenge for synthetic volume generation because the particles are sparse and display anisotropy in the clustering of particles as well as the individual particle shape. The sparseness and anisotropy make simulated annealing techniques[10] infeasible due to the complexity of calculating statistical descriptors that would be necessary to constrain the annealing process and capture the anisotropy of the structure. The sparse particle distribution also makes the decision tree machine learning model[23] ineffective because it depends on local information to build up a microstructure. Another challenge is the size of the training volumes. Here, the center training volume is 310 × 220 × 212 voxels, whereas the surface training volume is larger at 860 × 227 × 227. The generation method must be able to create unique volumes given the limited but differing amounts of training data. Data modulation methods such as mirroring or rotation cannot be used here because they may not preserve the anisotropy of the microstructure.

2.6 Evaluation

Quantification of the success of generation is not straightforward for the same reasons that it is difficult to recreate the microstructure through simulated annealing methods. First, a qualitative visual inspection is performed on volumes generated from the microstructure data. Next, 50 volumes are generated or sampled randomly from the training data set to perform a quantitative comparison. Both the training and generated volumes must be binarized before calculating statistics. To avoid a subjective choice that could influence statistics, the threshold value is chosen so that the binarized training data will have the same volume fraction as the original binary date before applying the down sampling operation. This procedure is performed for both sets of training data, and the same thresholds are then applied to the corresponding generated volumes. In addition, the volumes are filtered to remove any particles that are a single voxel. The volume fraction, particle size distribution, and particle shape distribution are compared in addition to two-point and clustering correlation functions to show how well the generated volumes resemble the training set. The two-point correlation is the probability that both ends of a vector randomly placed in the volume will fall on any particle. The clustering correlation is the probability that both ends of a vector randomly placed in the volume will fall on the same particle.

3. Results

The evolution of training the network for the center and surface regions is shown in Fig. 4, where each row shows a 2D slice of the volume viewed from the transverse direction created by the generator using the same initial input vector at different epochs during the training process. Initially, periodic features and repetitive patterns as well as some noise at the outer boundary can be seen at the earlier epochs. Both models progress towards a result that is qualitatively similar to the microstructure of the training dataset after training is completed at 300 epochs.

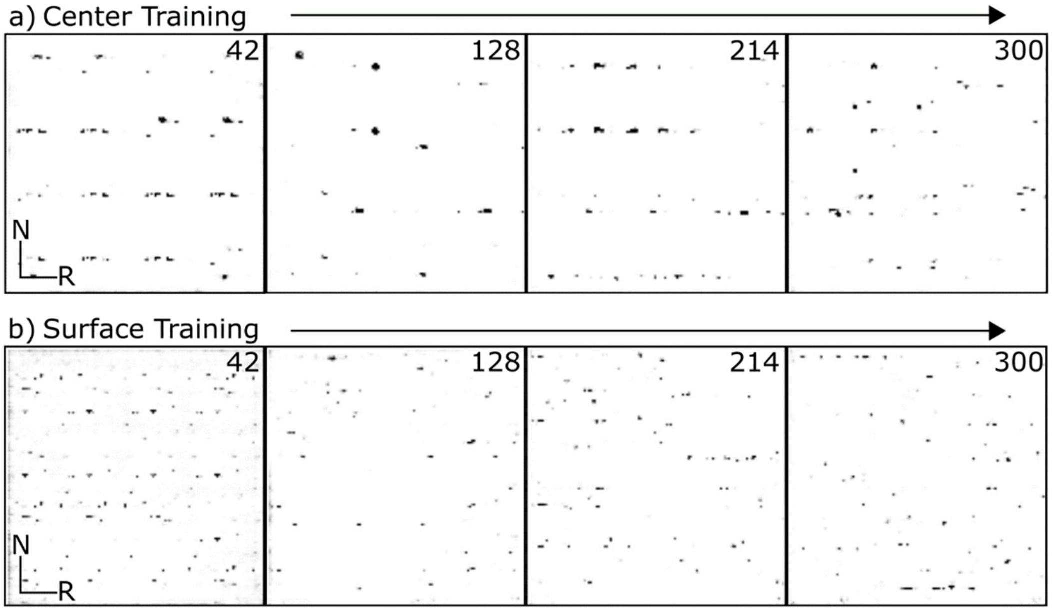

Fig. 4 2D slices of volumes that are generated at the labeled epoch for a a) center dataset and b) surface dataset

A qualitative comparison of three sections randomly selected in the training set and generated volumes as viewed from the normal, transverse, and rolling directions are shown in Fig. 5 for the center dataset and Fig. 6 for the surface dataset. Volumes created by the generator are visually representative of the training datasets with comparably sized particles and spatial distribution of particles. By viewing the data from different orientations, we demonstrate the generator's ability to capture anisotropic features of the datasets.

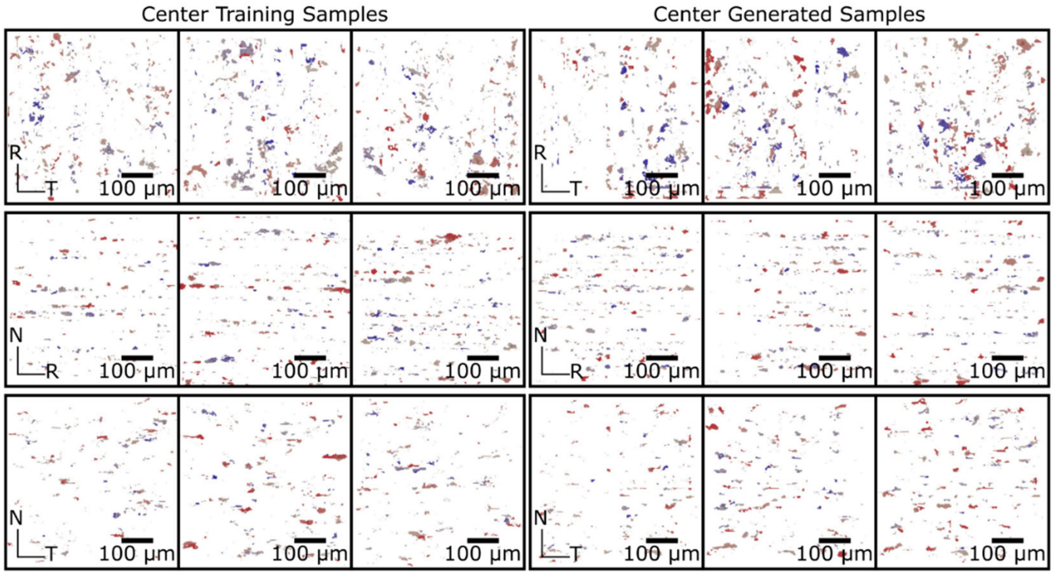

Fig. 5 Examples of randomly chosen sections taken from volumes of real and generated microstructures from the center datasets. Sections have a side length of 128 voxels (503 µm) and thickness of 50 voxels (196 µm) color coded based on depth through the section.

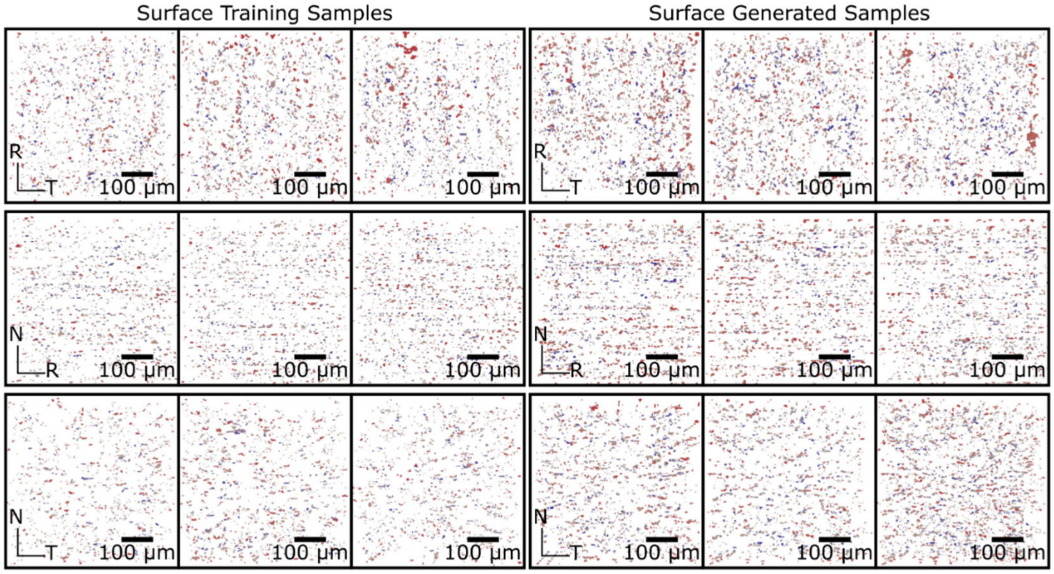

Fig. 6 Examples of randomly chosen sections taken from volumes of real and generated microstructures from the surface datasets. Sections have a side length of 128 voxels (503 µm) and thickness of 50 voxels (196 µm), color coded based on depth through the section.

A quantitative comparison between the real and generated volumes is provided in Table 3, where averaged characteristics are compared. The volume statistics include volume fraction and the number of particles within a sample and are calculated for each volume, and then the average and standard deviation are calculated over all 50 generated volumes. Particle statistics include the particle volume and particle length, which is defined as the largest length of the particle projected into the three primary axis directions. First, the average particle values are found for each volume. Next, the average and standard deviation of this value over all sampled volumes is calculated (Table 3). Generated volumes reasonably capture the features of the training data for both the surface and center datasets, although there is more variability within generated datasets. In particular, the number of particles within the volume for the center data has higher variability compared to other metrics. The size of particles is captured well based on the average values for particle volume and particle length.

Table 3 Average characteristics of the real and generated volumes

Dataset	Volume fraction (%)	Number of particles	Particle volume (μm^3)	Particle length (μm)		
				Transverse	Normal	Rolling
Surface Training	0.34±0.04	2117±207	204±4.2	7.0±0.12	5.3±0.08	7.6±0.13
Surface Generated	0.38±0.04	2169±188	219±5.7	7.2±0.11	5.5±0.06	8.0±0.12
Center Training	0.36±0.04	750±50	614±52.3	10.8±0.37	6.8±0.16	12.2±0.27
Center Generated	0.34±0.07	681±133	642±55.5	11.3±0.59	7.1±0.29	12.2±0.44

In addition to characteristics that have been averaged over the volume, it is important to ensure that the generated volumes capture the correct distribution of particle features. Kernel density estimation with a Gaussian kernel is used for a smooth estimate of the probability density distribution based on particle volume and length rather than fixed bins of a histogram. A distribution is estimated for each generated or training sample, and Fig. 7 shows the average distribution over all samples for the particle volume as well as particle lengths. The estimated probability density matches well between training and generated volumes for both datasets. Small differences are noted in the surface dataset where low volume particles are slightly less common compared to the training data. Similarly, the generated volumes capture the particle length distribution, and its anisotropy well compared to the training data for both the center and surface datasets.

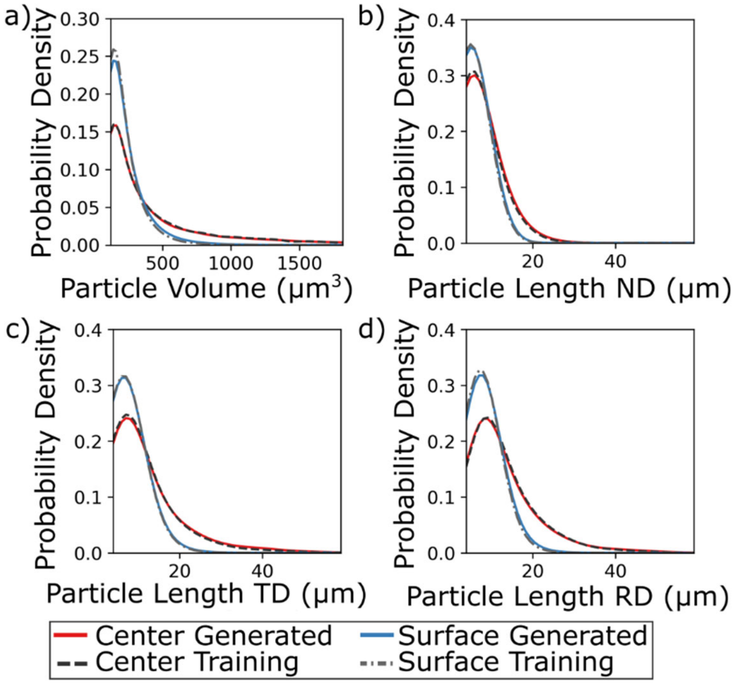

Fig. 7 Distribution of a) particle volume and particle length in the b) normal, c) transverse, and d) rolling directions

Figure 8 compares two-point and clustering correlation between training and generated samples for both data sets where data are simplified to show only the magnitude of the vector used in the correlation calculation. Generated samples show good agreement with training data and capture differences between the two data sets. The center dataset has a more gradual reduction in both correlation functions as a result of its larger particles. Clustering correlation is reduced to zero probability at distances greater than particle sizes in the samples, but two-point correlation levels out at a more constant value for longer range correlations. For this long-range, two-point correlation, the center and surface datasets have a similar value, which is captured by the generated samples.

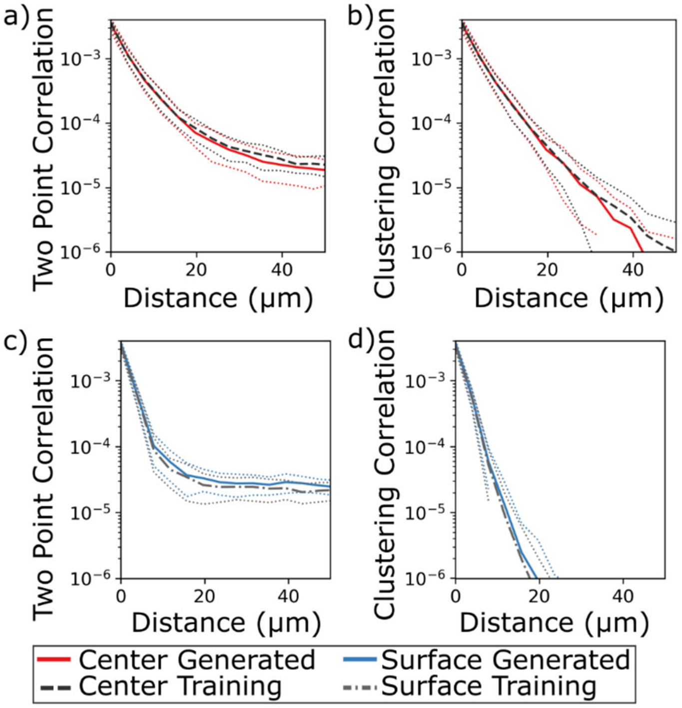

Fig. 8 a) Two-point and b) clustering correlation for the center dataset. c) Two-point and d) clustering correlation for the surface dataset. Dotted lines indicate standard deviation.

4. Discussion

The dataset presented here is a sparse distribution of particles clustered in an anisotropic pattern within the volume that would be difficult to capture through traditional means. The GAN approach with the random patch discriminator method has demonstrated the ability to generate unique 3D volumes that reasonably match the characteristics of this training data despite its limited size. The random patch discriminator improved stability during training to generate large volume sizes. Both the distribution of particle sizes, their anisotropy, and their spatial arrangement match the training data well, as confirmed by two-point and clustering correlation functions. The largest error in volume generation comes from the total number of particles, where there is greater variability in the generated volumes. However, the training dataset is within the range of variability for the generated volumes.

The GAN training instability is a problem that becomes progressively worse with larger volume output size. The use of random patches has improved training stability and enabled generation of volumes that statistically match the training set while also providing reasonable variability. By passing a minimal amount of information to the discriminator to classify the volume, we can reduce the number of trainable parameters to improve the training process. This approach is effective because it is possible to estimate the validity of a microstructure from a small sample without seeing the global volume. However, the size of images passed to the discriminator also presents a limitation. The success of the random patch method is dependent on the ability to capture all important characteristics of a material within the size of the image that is processed by the discriminator. Features in the dataset with a length scale greater than the discriminator input size would not be represented in generated volumes.

The random patch approach increases the possibility of using the number of patches as well as the size of the patch as input parameters. These parameters have been explored to determine their effect on the ability to train the network. First, a GAN is trained with a 1- and 10-patch discriminator for comparison with the 5-patch discriminator, which has been presented in previous sections. Here, all patches are cubes with a fixed edge length of 32 voxels. We found that by increasing the number of patches, we can reduce variability in the volume statistics, but a higher patch number did not significantly improve average results compared with the 5-patch discriminator. The 1-patch model produced more irregularly shaped particles compared with other models, particularly around the edges of the volumes, indicating that a 1-patch discriminator was not sufficient.

Next, the size of the patch is evaluated by comparing how well it represents the full data set. Figure 9 displays statistics for patches of size 16, 24, 32, 48, and 64 voxel edge length taken from training data to visualize where they become less representative compared to larger sized patches that better capture particle spatial distributions. Patch statistics show that training data in patches of size 24 voxel edge length or less begin to differ for volume fraction, number of particles, and particle volume compared to larger patches. These statistics also show reduced variability at greater patch sizes. This change in average value and variability indicates where patches may not effectively capture the features of the full data set. This initial evaluation before training is useful to help estimate an effective patch size, which will vary based on the dataset with which it will be used. A patch that is not of a sufficient size to capture long-range features of the microstructure is unlikely to be effective for use in a discriminator to evaluate the validity of a given sample.

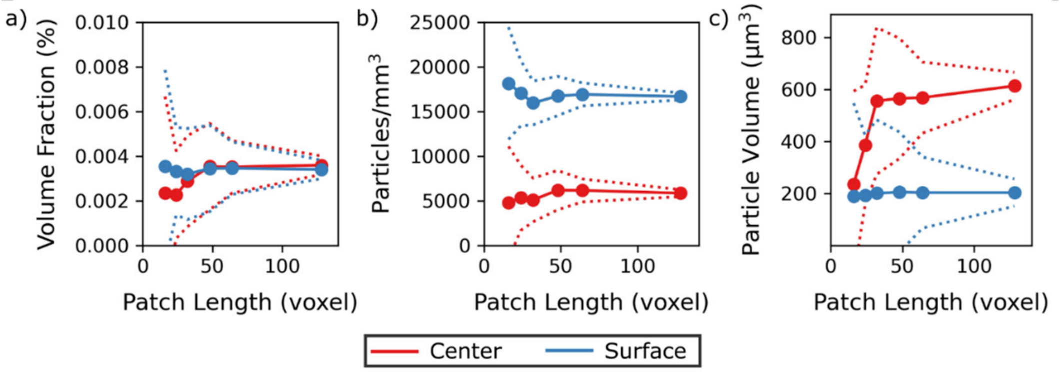

Fig. 9 Patch statistics for a) volume fraction, b) particles/mm^3, and c) average particle volume calculated from patches between 16 and 128 voxel edge length for the training datasets. Dotted lines indicate standard deviation.

Next, training of a GAN is performed with patch sizes of 24, 48, and 64 voxel edge length to verify performance using different patch sizes. Each of these networks uses five patches in the discriminator. The 24-voxel cube patch was able to generate microstructures that were qualitatively similar to the training data using visual inspection, but it contained more repetitive particle shapes and distribution within each individual volume similar to the early training steps in Fig. 4. This reduced performance indicates that the patch size was able to partially represent the dataset, but it did not fully capture the long-range connections needed to allow for generation of realistic clusters of particles. We expect that even smaller patches would perform poorly based on the patch statistics (see Fig. 9). By increasing the patch size, there is less variability between volumes, with similar particles appearing at the same location in multiple generated volumes. This issue could potentially be resolved by obtaining a larger quantity of unique training data, compared to that which is used here. However, the ability to train the GAN with a limited training set is an advantage of the smaller random patch sizes. In addition, the larger patch sizes did not show a significant improvement in average volume statistics compared to the 32-voxel patch, which also requires greater memory and training time. Therefore, the smallest patch size that captures the features of the particle distribution is the most efficient in terms of both training and data collection requirements.

Many GAN architectures have been developed, the majority of which have been applied to problems involving 2D images. It is worth examining differences between this work and other GAN architectures that have been applied to 3D microstructure reconstruction. Hsu et al.[29] developed a Wasserstein GAN (WGAN)[37] with spectral normalization[38] for use with 3D microstructure data with a volume size of 95 × 95 × 95 voxels. This network architecture was implemented and applied to two different two-phase microstructure generation problems (shown in Fig. 10). Figure 10, left panel, shows training and generated data for a microstructure with approximately 50% volume fraction of each phase, whereas Fig. 10, right panel, shows training and generated data for the sparse second-phase particle microstructure studied in this work. Although this WGAN was able to successfully generate 3D volumes with a more balanced volume fraction, it did not perform as well for the second-phase microstructure. Here, the generated results have some poorly shaped particles and less variability in the output where particles of the same size have been generated in the same position in multiple volumes. This reduced performance could be the result of the initially larger convolutions used on the smaller particles or because of the relatively limited size of the training data in the center training volume that we used. In contrast, our random patch discriminator was able to generate varied volumes even with these constraints. In addition, we expect that by increasing the size of the generated volume, performance will be reduced. In contrast, in our method, we detach the size of the generator and discriminator and prevent challenges that occur when using a large discriminator network.

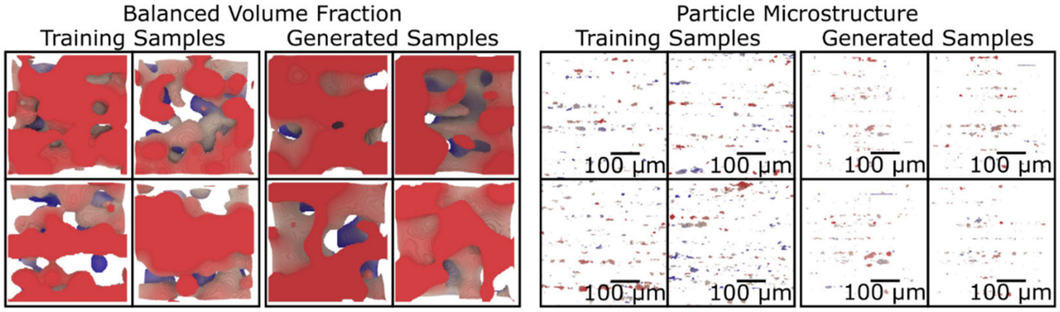

Fig. 10 Results from a WGAN based on Hsu et al.[29] demonstrating good performance for a dataset with balanced volume fraction similar to that which the network was originally developed for. Results when applied to a second-phase particle microstructure show lower performance compared with the random patch method.

Another alternative approach is a GAN with residual blocks rather than standard convolutions,[38,39] which has shown high performance in GAN models for 2D images. A network was developed with a ResNet[40]-based discriminator that used spectral normalization and hinge loss[41,42] and was applied to the second-phase particle microstructure. Again, this network was unable to reproduce the details of the second-phase microstructure when both the generator and discriminator are designed for a 128-voxel edge length cube. Next, the random patch method was applied to this discriminator with residual blocks. This network was able to successfully generate second-phase microstructures that reasonably captured the characteristics of the training data. However, it did not show improvement over the original random patch approach while requiring significantly more training time.

5. Conclusion

In this study, we describe a GAN approach that uses a random patch discriminator method that enabled stable training and produced volumes of 128 × 128 × 128 voxels based on second-phase particle microstructure training data. The results demonstrate some unique challenges for applying a GAN to 3D volumes and particularly to second-phase particle microstructures. Our random patch discriminator enabled stable training of the GAN despite a limited set of training data as well as challenging sparse anisotropic microstructural features. The generated volumes are visually similar to the training data, and they also capture a similar statistical distribution of particle characteristics. After training, the generator model can quickly produce a large quantity of statistically representative, synthetic microstructures to be used to evaluate material models and provide a better understanding about how variability in the particle microstructure contributes of failure.

6. References

1. Brocks W, Sun D-Z, Hönig A. Verification of the transferability of micromechanical parameters by cell model calculations with visco-plastic materials. Int J Plast. 1995;11(8):971–989.

2. Broek D. The role of inclusions in ductile fracture and fracture toughness. Eng Fract Mech. 1973;5(1):55–66.

3. Garrett G, Knott J. The influence of compositional and microstructural variations on the mechanism of static fracture in aluminum alloys. Metall Mater Trans A. 1978;9(9):1187–1201.

4. Van Stone R, Psioda J. Discussion of "metallurgical factors affecting fracture toughness of aluminum alloys". Metall Mater Trans A. 1975;6(4):668.

5. Woodward R. Penetration behaviour of a high-strength aluminum alloy. Metals Technology. 1979;6(1):106–110.

6. Gologanu M, Leblond J-B, Devaux J. Approximate models for ductile metals containing non-spherical voids—case of axisymmetric prolate ellipsoidal cavities. J Mech Phys Solids. 1993;41(11):1723–1754.

7. Koplik J, Needleman A. Void growth and coalescence in porous plastic solids. Int J Solids Struct. 1988;24(8):835–853.

8. Pardoen T, Hutchinson J. An extended model for void growth and coalescence. J Mech Phys Solids. 2000;48(12):2467–2512.

9. Jiao Y, Stillinger F, Torquato S. A superior descriptor of random textures and its predictive capacity. PNAS. 2009;106(42):17634–17639.

10. Yeong C, Torquato S. Reconstructing random media. Phys Rev E. 1998;57(1):495.

11. Čapek P, Hejtmánek V, Brabec L, Zikánová A, Kočiřík M. Stochastic reconstruction of particulate media using simulated annealing: improving pore connectivity. Transp Porous Media. 2009;76(2):179–198.

12. Gerke K, Karsanina M, Skvortsova E. Description and reconstruction of the soil pore space using correlation functions. Eurasian Soil Sci. 2012;45(9):861–872.

13. Manwart C, Torquato S, Hilfer R. Stochastic reconstruction of sandstones. Phys Rev E. 2000;62(1):893.

14. Rintoul MD, Torquato S. Reconstruction of the structure of dispersions. J Colloid Interface Science. 1997;186(2):467–476.

15. Chellappa R, Chatterjee S, Bagdazian R. Texture synthesis and compression using Gaussian-Markov random field models. IEEE Trans Syst Man Cybern. 1985(2):298–303.

16. Efros AA, Leung TK, editors. Texture synthesis by non-parametric sampling. Proceedings of the Seventh IEEE International Conference on Computer Vision; 1999 Sep 20–27; Kerkyra, Greece. IEEE.

17. Levina E, Bickel PJ. Texture synthesis and nonparametric resampling of random fields. Ann Stat. 2006;34(4):1751–1773.

18. Paget R, Longstaff ID. Texture synthesis via a noncausal nonparametric multiscale Markov random field. IEEE Trans Image Process. 1998;7(6):925–931.

19. Javaheri I, Sundararaghavan V. Polycrystalline microstructure reconstruction using Markov random fields and histogram matching. Comp Aided Des. 2020;120:102806.

20. Kumar A, Nguyen L, DeGraef M, Sundararaghavan V. A Markov random field approach for microstructure synthesis. Model Simul Mater Sci Eng. 2016;24(3):035015.

21. Liu X, Shapiro V. Random heterogeneous materials via texture synthesis. Comput Mater Sci. 2015;99:177–189.

22. Turner DM, Kalidindi SR. Statistical construction of 3-D microstructures from 2-D exemplars collected on oblique sections. Acta Mater. 2016;102:136–148.

23. Bostanabad R, Chen W, Apley D. Characterization and reconstruction of 3D stochastic microstructures via supervised learning. J Microsc. 2016;264(3):282–297.

24. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial nets. Adv Neural Inf Process. 2014;27.

25. Li X, Yang Z, Brinson LC, Choudhary A, Agrawal A, Chen W, editors. A deep adversarial learning methodology for designing microstructural material systems. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference; 2018 Aug 26–29; Quebec City, Canada. American Society of Mechanical Engineers.

26. Cang R, Li H, Yao H, Jiao Y, Ren Y. Improving direct physical properties prediction of heterogeneous materials from imaging data via convolutional neural network and a morphology-aware generative model. Comput Mater Sci. 2018 Jul 1;150:212–221.

27. Singh R, Shah V, Pokuri B, Sarkar S, Ganapathysubramanian B, Hegde C. Physics-aware deep generative models for creating synthetic microstructures. arXiv. 2018 Nov 21. arXiv:1811.09669. https://arxiv.org/abs/1811.09669.

28. Mosser L, Dubrule O, Blunt MJ. Reconstruction of three-dimensional porous media using generative adversarial neural networks. Phys Rev E. 2017;96(4):043309.

29. Hsu T, Epting WK, Kim H, Abernathy HW, Hackett GA, Rollett AD, Salvador PA, Holm EA. Microstructure generation via generative adversarial network for heterogeneous, topologically complex 3D materials. JOM. 2021 Jan;73(1):90–102.

30. Gayon-Lombardo A, Mosser L, Brandon NP, Cooper SJ. Pores for thought: Generative adversarial networks for stochastic reconstruction of 3D multi-phase electrode microstructures with periodic boundaries. Npj Comput Mater. 2020 Jun 25;6:82.

31. Zhang F, Teng Q, Chen H, He X, Dong X. Slice-to-voxel stochastic reconstructions on porous media with hybrid deep generative model. Comput Mater Sci. 2021;186:110018.

32. Brock A, Donahue J, Simonyan K. Large scale GAN training for high fidelity natural image synthesis. 2018. arXiv. arXiv:180911096. https://arxiv.org/abs/1809.11096.

33. Karras T, Laine S, Aittala M, Hellsten J, Lehtinen J, Aila T, editors. Analyzing and improving the image quality of stylegan. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020 June 13–19; Seattle, WA. IEEE.

34. Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks. 2015. arXiv. arXiv:151106434. https://arxiv.org/abs/1511.06434.

35. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, Desmaison A. Pytorch: An imperative style, high-performance deep learning library. Adv Neural Inf Process. 2019;32.

36. Kingma DP, Ba J. Adam. A method for stochastic optimization. 2014 Dec 22. arXiv. arXiv:1412.6980.

37. Arjovsky M, Chintala S, Bottou L. Wasserstein generative adversarial networks. International Conference on Machine Learning. 2017 Jul 6–11. Sydney, NSW, Australia. PMLR; pp. 214–223.

38. Miyato T, Kataoka T, Koyama M, Yoshida Y. Spectral normalization for generative adversarial networks. 2018 Feb 16. arXiv. arXiv:1802.05957. https://arxiv.org/abs/1802.05957.

39. Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V, Courville AC. Improved training of Wasserstein GANs. Adv Neural Inf Process. 2017;30.

40. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016 June 27–30. Las Vegas, NV. IEEE pp. 770–778.

41. Lim JH, Ye JC. Geometric GAN. 2017 May 8. arXiv. arXiv:1705.02894. https://arxiv.org/abs/1705.02894.

42. Tran D, Ranganath R, Blei D. Hierarchical implicit models and likelihood-free variational inference. Adv Neural Inf Process. 2017;30.

List of Symbols, Abbreviations, and Acronyms

2D	two-dimensional
3D	three-dimensional
CT	X-ray computed tomography
DCGAN	deep convolution GAN
GAN	generative adversarial network
NA	not applicable
ReLU	rectified linear unit
WGAN	Wasserstein GAN

1 DEFENSE TECHNICAL
(PDF) INFORMATION CTR
DTIC OCA

1 DEVCOM ARL
(PDF) FCDD RLB CI
TECH LIB

40 DEVCOM ARL
(PDF) FCDD RLA A
S KARNA
J NEWILL
A RAWLETT
S SCHOENFELD
FCDD RLA C
P GILLICH
FCDD RLA T
R FRANKART
FCDD RLW TA
S BILYK
FCDD RLA TB
S ALEXANDER
R BANTON
T BAUMER
D CASEM
J CLAYTON
B FAGAN
J GARVEY
A GOERTZ
A GUNNARSSON
C HAMPTON
R KARGUS
D KRAYTERMAN
M KLEINBERGER
E MATHEIS
J MCDONALD
P MCKEE
C MEREDITH
T PLAISTED
K RAFAELS
S SATAPATHY
L SHANNAHAN
M TEGTMEYER
C WEAVER
T WEERASOORIYA
S WOZNIAK
T ZHANG
FCDD RLA TG
N GNIAZDOWSKI
FCDD RLA TD
B KRZEWINSKI
FCDD RLA TE
J LLOYD
P SWOBODA
G VUNNI
FCDD RLA MB
D MAGAGNOSC
Z WILSON