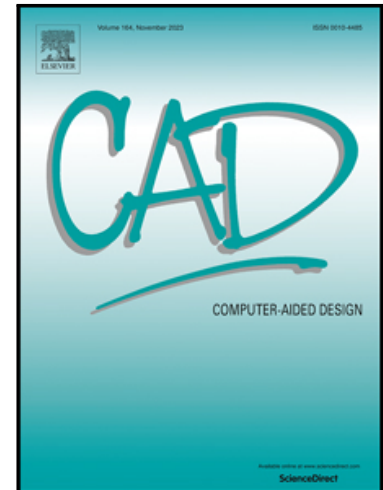


3D stochastic microstructure reconstruction via slice images and attention-mechanism-based GAN

Ting Zhang , Ningjie Bian , Xue Li

PII: S0010-4485(24)00087-3  
DOI: <https://doi.org/10.1016/j.cad.2024.103760>  
Reference: JCAD 103760



To appear in: *Computer-Aided Design*

Received date: 21 March 2024  
Revised date: 5 June 2024  
Accepted date: 28 June 2024

Please cite this article as: Ting Zhang , Ningjie Bian , Xue Li , 3D stochastic microstructure reconstruction via slice images and attention-mechanism-based GAN, *Computer-Aided Design* (2024), doi: <https://doi.org/10.1016/j.cad.2024.103760>

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

**Highlights**

- 2D slices of a 3D image are used as training images to lower GPU burdens.
- The convolutional triplet attention is used to prioritize the learned features.
- Only a 3D image is required for whole training.

Journal Pre-proof

### 3D stochastic microstructure reconstruction via slice images and attention-mechanism-based GAN

Ting Zhang<sup>1</sup>, Ningjie Bian<sup>1</sup>, Xue Li<sup>2, \*</sup>

(\*corresponding author, E-mail: lixue@dicp.ac.cn)

1. College of Computer Science and Technology, Shanghai University of Electric Power, Shanghai 200090, China

2. National Engineering Research Center of Lower-Carbon Catalysis Technology, Dalian Institute of Chemical Physics, Chinese Academy of Sciences, Dalian 116023, China

**Abstract:** Stochastic media are used to characterize materials with irregular structure and spatial randomness, and the remarkable macroscopic features of stochastic media are often determined by their internal microstructure. Hardware loads and computational burdens have always been a challenge for the reconstruction of large-volume materials. To tackle the aforementioned concerns, this paper proposes a learning model based on generative adversarial network that uses multiple 2D slice images to reconstruct 3D stochastic microstructures. The whole model training process requires only a 3D image of stochastic media as the training image. In addition, the attention mechanism captures cross-dimensional interactions to prioritize the learned features and improves the effectiveness of training. The model is tested on stochastic porous media with two-phase internal structure and complex morphology. The experimental findings demonstrate that utilizing multiple 2D images helps the model learn better and reduces the occurrence of overfitting, while greatly reducing the hardware loads of the model.

**Key words:** stochastic media; attention mechanism; generative adversarial network; microstructure; cross-dimensional interaction

#### 1. Introduction

Nowadays, stochastic media are widely used and exist in the world, but their properties are quite different even their outside surfaces look quite similar. One of the most important reasons is that the performance of these media is largely influenced by their internal microstructures, so studying the internal microstructures of these media holds significant importance. A large portion of these media has stochastic shapes and irregular pores in their inside microstructures, which cannot be directly described through straightforward equations or languages. Therefore, the internal curves and shapes bring over many issues such as randomness and anisotropy, challenging the design of such stochastic media in the engineering and scientific fields. Therefore, the problem of obtaining the digital 3D structures of stochastic media has been raised. For the reconstruction or simulation of microstructures in stochastic media, the two principal categories of reconstruction

methods, physical experiments and numerical reconstruction, were developed [1].

The basic principle of the physical experiment method entails utilizing a high-precision scanning instrument for scanning samples in order to acquire 2D cross-sectional data, and then sequentially overlay these 2D images into a 3D model after post-processing these data. The representative physical experiment methods comprise the serial sections tomography methods (SSTM) [2], the focused ion beam (FIB) [3], and the CT [4]. However, the hardware cost of these methods is too high and the physical manufacturing process is quite complicated, which leads to easy damage to the internal structure of the sample, thus limiting the suitability of the physical experiment methods.

When contrasted with physical experimental methods, numerical reconstruction methods employ a substantial quantity of 2D thin slices or 3D microstructures as training images (TIs) and utilize the statistical information obtained from these images to reconstruct microstructures. There are likewise many techniques for numerical reconstruction, such as the simulated annealing method [5], the Markov chain Monte Carlo method [6], and multi-point statistics (MPS) [7]. Numerical reconstruction methods, compared to physical experimental methods, offer greater cost efficiency and have the ability of reconstructing multiple stochastic outcomes of microstructures. However, the numerical reconstruction methods also have their respective limitations. For example, the simulated annealing method solely relies on low-order statistical data within the image, and thus it is difficult to reproduce some high-order properties (such as the correlations among multiple neighboring pixels/voxels) in the microstructures. Another example is MPS. Although MPS has been widely adopted and many variants have been developed, it still consumes a considerable amount of time, even with the help of GPUs [8].

With the increasing computing power of graphic cards and thanks to the strong learning ability of deep learning, the scientific and engineering problems for stochastic reconstruction or simulation that can be solved by deep learning and neural networks have increased dramatically. Goodfellow et al. [9] introduced the concept of generative adversarial networks (GAN), triggering a great deal of research work to generate stochastic models conforming to the distribution of training data. Since GAN theoretically can fit any distributions, it helps to describe the complicated stochastic shape and curves in the internal microstructure of media. Mosser et al. [10] initially employed GAN to reconstruct stochastic media. Subsequently, GAN and its derivatives have been increasingly extensively utilized for stochastic reconstruction of media. Volkhonskiy et al. [11] successfully reconstructed 3D media from 2D center slices by combining the autoencoder with a DCGAN model. Shams et al. [12] addressed the issue of reconstructing 3D porous rocks by combining DCGAN with the autoencoder. Evgeniy et al. [13] utilized VAE-GAN to reconstruct 3D stochastic media from 2D images and omitted the element-wise descriptor. Zhang et al. [14, 15]

employed multi-stage GAN approach to reconstruct 3D stochastic media.

Humans are able to quickly discern key information in complex data when they see it because humans are able to quickly discern key information while ignoring certain secondary details from complex data. When extracting features from TIs, traditional deep learning methods almost all have the disadvantage that it is often difficult to focus on key features or suppress irrelevant information in complex training data. In order to deal with this problem, the attention mechanism (AM) has been proposed and extensively employed in diverse deep learning engineering models to understand the significance of key information in the environment by concentrating on it [16-18].

Based on SinGAN-- a classical single-image generation model [19], Hinz et al. [20] proposed a refined model ConSinGAN, which uses concurrent training and a multi-staged approach to improve the training and generation capabilities. Although ConSinGAN can generate multi-staged images, its corresponding real training data is only a single image, which is easily prone to overfitting. However, if multiple TIs are used, the increase in memory and training time becomes an obvious challenge.

Therefore, to overcome the above difficulties, this paper proposes a model for the reconstruction of 3D stochastic microstructures of media, which is called slice-image and attention-mechanism-based GAN (SAGAN), integrating GAN with an AM module referred to as the convolutional triplet attention module (CTAM) [21]. A single 3D TI of stochastic media is sliced before training and then divided into three real datasets with three different directions (X, Y and Z). The input image to the discriminator is neither the whole 3D image nor a single 2D cross-section, but multiple 2D slices, which significantly reduces the GPU memory footprint, as 2D images have a lower GPU memory load in the training process. In addition, the selected CTAM enables cross-dimensional interactions and prioritization of the key attributes to improve the reconstruction results. Finally, we compare SAGAN with different methods and training data to prove its effectiveness.

## 2. Methodology

### 2.1 CTAM

To maximize the utilization of the restricted visual data, humans consciously focus on the important information while disregarding certain secondary details, which is the basic principle of AM. Currently, AM has become a crucial concept in neural networks and an effective method for capturing the significance of individual local details especially within the image data. This technique finds extensive application across various deep learning methods, especially when dealing with features that are difficult to recognize, and has significant advantages. Its definition

usually involves the use of specific mathematical formulas or terms [22]. A CTAM comprises three parallel branches, with two tasked with capturing cross-dimensional interaction between the channel dimension  $C$  and either the spatial dimension  $W$  or  $H$  ( $W$  is width;  $H$  is height). The third branch constructs spatial attention. The output from all three branches is then aggregated through averaging.

### 2.1.1 Cross-dimensional interaction

The purpose of the cross-dimensional interaction is to capture the interplay between the spatial and channel dimensions of the input data. Utilizing cross-dimensional interaction in triplet attention entails assigning three branches to capture dependency relationships within the  $(C, W)$ ,  $(C, H)$  and  $(W, H)$  dimensions of the input data, respectively. The specific implementation of each branch is specified in Section 2.1.3.

### 2.1.2 Z-pool

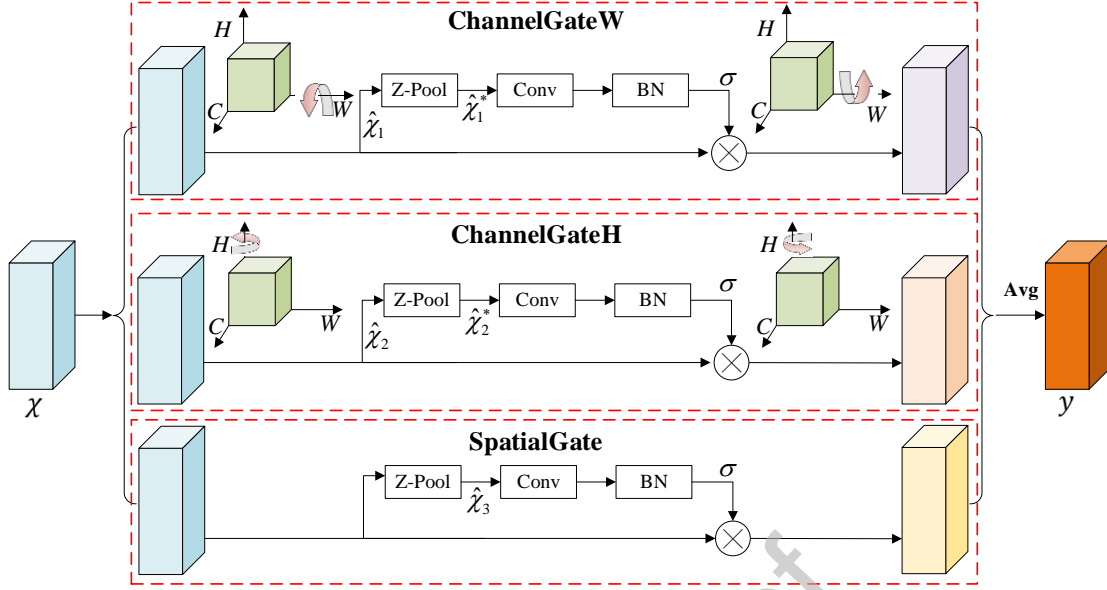
The Z-pooling layer converts the first dimension of input from  $C$  to 2 by combining the average and max pooled characteristics, which allows the layer to preserve a rich representation of the actual data while decreasing its depth, thus simplifying subsequent computation, expressed as follows:

$$Z - pool(\chi) = [MaxPool_{0d}(\chi), AvgPool_{0d}(\chi)], \quad (1)$$

where  $\chi$  is the input;  $MaxPool_{0d}(\chi)$  is the operation of max-pooling the first dimension of  $\chi$ ;  $AvgPool_{0d}(\chi)$  is the operation of average-pooling the first dimension of  $\chi$ . Finally, the  $Z - pool$  of the data with the shape  $(C \times W \times H)$  generates the data with the shape  $(2 \times W \times H)$ .

### 2.1.3 Triplet attention

Given the above information, triplet attention can be described as a module with three branches that receives input data and produces refined data with the same dimensions, as illustrated in Fig. 1.



**Fig. 1.** The structure of CTAM. It has three branches (ChannelGateW, ChannelGateH and SpatialGate), the first two of which (ChannelGateW and ChannelGateH) establish the connection between  $C$  and either  $W$  or  $H$  using rotation. The third one SpatialGate is employed to capture the relationship of spatial dependency ( $W$  and  $H$ ) and finally an average is used to aggregate the weights.  $\otimes$  represents the element-wise multiplication.  $\sigma$  denotes the sigmoid activation function.

To simplify the comprehension and explanation of CTAM, we choose 2D perspectives over 3D ones. Suppose the input and output images in Fig. 1 are all 2D images. Given input data  $\chi \in R^{C \times W \times H}$ , it is then fed into each of the three branches. The cross-dimensional interaction of  $C$  with  $W$  is established in the first branch named ChannelGateW. To achieve this, the input  $\chi$  is rotated 90° counterclockwise along the  $W$  axis and obtain the rotated data named  $\hat{\chi}_1$  which is in the shape  $(H \times W \times C)$ . Then  $\hat{\chi}_1^*$  (dimension:  $2 \times W \times C$ ) is obtained from  $\hat{\chi}_1$  by passing through Z-pool. Next  $\hat{\chi}_1^*$  undergoes a standard convolutional layer (called “Conv”, with a kernel size of  $k \times k$ ) followed by a batch normalization layer (called “BN”), yielding an intermediate output with the dimension  $(1 \times W \times C)$ . The resulting attention weights are derived by passing the above intermediate output through a sigmoid activation function ( $\sigma$ ). The generated attention weights and  $\hat{\chi}_1$  are then multiplied element-wisely (represented by “ $\otimes$ ”) and then rotated 90° clockwise along the  $W$  axis to return to the initial shape of  $\chi$ . Likewise, the cross-dimensional interaction of  $H$  with  $C$  is established in the second branch named ChannelGateH. To realize the interaction between spatial ( $W$  and  $H$ ) and channel ( $C$ ) dimensions, note that  $C$  should not be set to 1.

The final branch named SpatialGate is employed to construct spatial attention. The input  $\chi$

first remains unchanged. Then the channels of  $\chi$  are changed to two by the  $Z - pool$ , and subsequently this  $(2 \times W \times H)$  processing data named  $\hat{\chi}_3$  is input to the third branch to produce the attention weights with the dimension  $(1 \times W \times H)$  which are subsequently element-wisely multiplied by  $\chi$ .

Finally, the output of the three branches  $(C \times W \times H)$  is combined through averaging. The process of obtaining the refinement data  $y$  from input data  $\chi \in R^{C \times W \times H}$  using CTAM can be defined as:

$$y = \frac{1}{3} \left( \overline{\hat{\chi}_1 \sigma(bn(\psi_1(\hat{\chi}_1^*)))} + \overline{\hat{\chi}_2 \sigma(bn(\psi_2(\hat{\chi}_2^*)))} + \chi \sigma(bn(\psi_3(\hat{\chi}_3))) \right), \quad (2)$$

where  $\sigma(\cdot)$  denotes the sigmoid activation function;  $\psi_1(\cdot)$ ,  $\psi_2(\cdot)$ , and  $\psi_3(\cdot)$  refer to the standard convolutional layers in the three branches of CTAM;  $bn(\cdot)$  denotes the batch normalization layer;  $\hat{\chi}_1$  and  $\hat{\chi}_2$  are respectively obtained by rotating  $\chi$  counterclockwise by  $90^\circ$  along the  $W$  and  $H$  axes;  $\hat{\chi}_1^*$ ,  $\hat{\chi}_2^*$ , and  $\hat{\chi}_3$  are respectively obtained from  $\hat{\chi}_1$ ,  $\hat{\chi}_2$ , and  $\chi$  by passing through  $Z - pool$ ; overbars mean the  $90^\circ$  clockwise rotation.

To simplify Eq. (2),  $y$  becomes:

$$y = \frac{1}{3} (\overline{\hat{\chi}_1 \omega_1} + \overline{\hat{\chi}_2 \omega_2} + \chi \omega_3), \quad (3)$$

where  $\omega_1$ ,  $\omega_2$  and  $\omega_3$  refer to the attention weights.

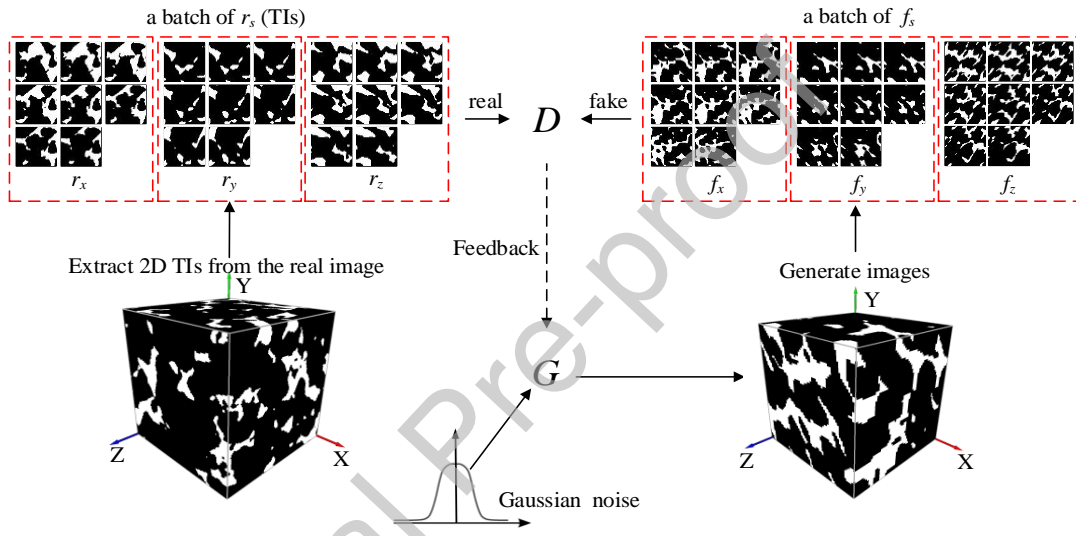
## 2.2 Preparation of training data

In GAN, random noise is input to the generator (called “ $G$ ”) and real training data are input to the discriminator (called “ $D$ ”). Having a sufficient amount of data as the real input to  $D$  can often provide a more detailed description of the stochastic microstructures of media, but the more detailed or larger the size of the data, the more likely the training time is too long and the GPU burdens are too high. If a single 2D image is used as the real input to the  $D$ , the vertical z-axis statistical information is not available. However, if a whole 3D image is used as the TI, the computational burden and memory occupation in GPU calculation can be serious. Therefore, we use multiple 2D images as input for  $D$  by slicing a real 3D image into a 2D image sequence along one direction. Note that the 3D image is sliced along three axes and therefore three real datasets are obtained along three directions. We also slice the 3D fake image generated by  $G$  before it is fed into the  $D$ . The training process and preparation of training data of SAGAN is proposed hereafter.

To ensure that the model fully learns the features in each direction during training, in each epoch,  $D$  and  $G$  are trained using the 2D TIs sliced along the X, Y and Z directions. Suppose  $z$  is Gaussian noise and let  $G(z)$  represent generated images from the  $G$ . Let  $s = x, y$ , and  $z$  respectively corresponding to the X, Y, and Z directions of 3D space. To train  $D$ , a batch of  $f_s$  (fake 2D slices, and batch size = 8) are sampled from  $G(z)$  along the direction  $s$  (= X, Y, and Z



respectively), and a batch of  $r_s$  (real 2D slices, and batch size = 8) are sampled from the real image along the direction  $s$ .  $f_x$ ,  $f_y$ , and  $f_z$  are the images sampled from  $G(z)$  along the X, Y and Z directions respectively. Similarly,  $r_x$ ,  $r_y$ , and  $r_z$  are the images sampled from the real 3D image along the X, Y and Z directions respectively. Then, they are fed into  $D$  together for training. Note that both  $r_s$  and  $f_s$  from the same direction must be cooperatively used, meaning for  $r_x$  and  $f_x$  (2D slices of X direction),  $r_y$  and  $f_y$  (2D slices of Y direction),  $r_z$  and  $f_z$  (2D slices of Z direction), these three datasets must be cooperatively used.  $D$  feeds the results back to  $G$ , making  $G$  produce a more realistic image. Fig. 2 shows an overview training process and preparation of training data of SAGAN.

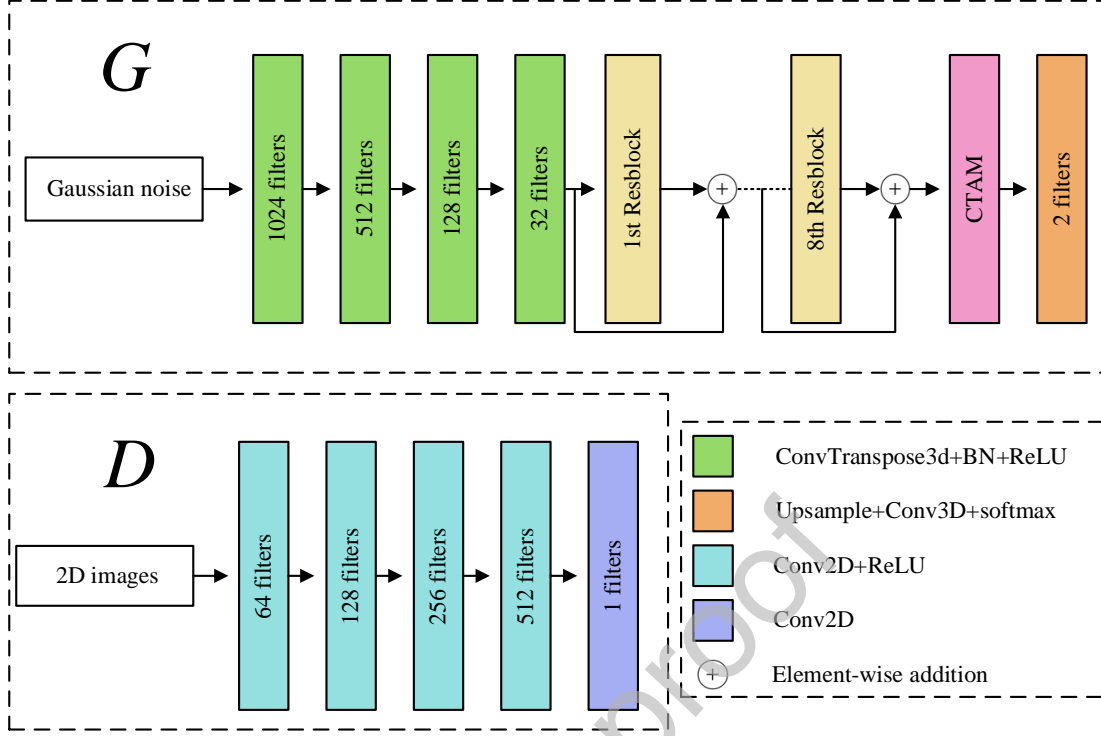


**Fig. 2.** An overview training process and preparation of training data of SAGAN. The generator produces 3D two-phase images of stochastic media, which are then sliced along each of the three axes, obtaining three collections of 2D slices that serve as input for the discriminator. The 2D slice TIs with the same direction are utilized as the real input for the discriminator. A batch of  $f_s$  (or  $r_s$ ) contains 8 images (batch size = 8).  $r_s$  and  $f_s$  of the same direction must be cooperatively used.

### 2.3 The structure of SAGAN

The details of  $D$  and  $G$  are shown in Fig. 3. In our design, there are four ConvTranspose3d layers, eight residual blocks (Resblocks), one CTAM and one 3D convolutional layer in  $G$ ; there are five 2D convolutional layers in  $D$ . Resblocks can mitigate the likelihood of deterioration. CTAM can assist the model to concentrate on the significant features of the TI. Gaussian noise is used as the input of  $G$  and its shape is  $8 \times 32 \times 4 \times 4 \times 4$  (batch size: 8; channel: 32; depth, width and height are all 4), which can be viewed as 256 ( $=8 \times 32$ ) 3D Gaussian noise cubes ( $4 \times 4 \times 4$  each). The input 2D images of  $D$  is  $8 \times 2 \times 64 \times 64$  (batch size: 8; nPhase: 2; width and height are both 64). Note that “nPhase” means the number of phases in the images. For binary images, nPhase = 2; for

$n$ -phase materials,  $nPhase = n$ . Upsampling is used in the last two filters of  $G$ .



**Fig. 3.** The architecture of SAGAN.  $G$  consists of four ConvTranspose3d layers, 8 Resblocks, 1 CTAM module, and one 3D convolutional layer. The hyperparameters of the ConvTranspose3d layers are kernel=4×4×4, stride=2, padding=2. The 3D convolutional layer is kernel=3×3×3, stride=1.  $D$  consists of four 2D convolutional layers with ReLU and one 2D convolutional layer. The hyperparameters of the first four 2D layers are kernel=4×4, stride=2, padding=1; the last 2D layer is kernel=4×4, stride=2.

#### 2.4 The loss function of SAGAN

The original GAN suffers from training instability and mode collapse. To solve this problem, we choose WGAN-GP as the loss function. WGAN-GP uses Wasserstein distance to quantify the disparity between the distribution of generated data and that of real data [23], thus keeping the model stable. The loss is formulated as:

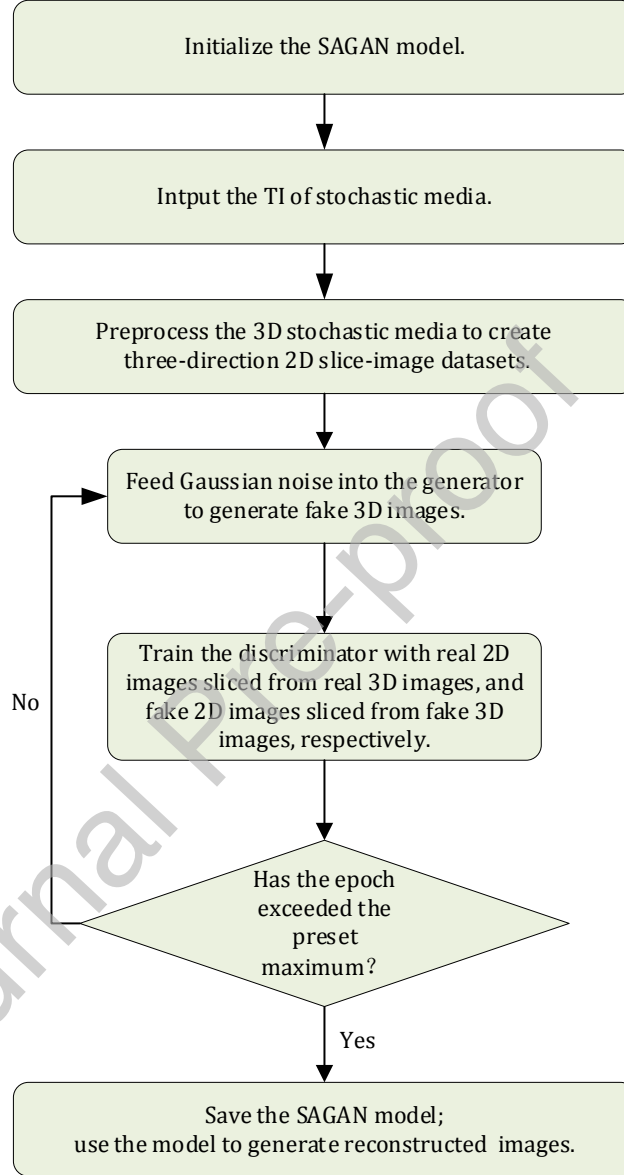
$$L(G, D) = E_{f_s \sim P_g} [D(f_s)] - E_{r_s \sim P_r} [D(r_s)] + \lambda E_{k \sim P_k} [(\|\nabla_k D(k)\|_2 - 1)^2], \quad (4)$$

where  $P_r$  is the real data distribution and  $P_g$  is the generated data distribution;  $f_s$  and  $r_s$  ( $s = x, y, z$ ) respectively represent the fake 2D slices from the generated 3D fake images and the real 2D slices in the X, Y and Z directions;  $k = \epsilon r_s + (1 - \epsilon) * f_s$ , where  $\epsilon \in [0, 1]$ ;  $P_k$  is the distribution of the interpolated samples;  $k \sim P_k$  involves sampling points uniformly along straight lines between pairs of sampled points from  $P_r$  and  $P_g$ ;  $\nabla_k D(k)$  is the gradient with respect to  $k$ .  $\lambda$  is the weight, set to 10 by default. As mentioned previously, " $r_x$  and  $f_x$ ", " $r_y$  and  $f_y$ ", " $r_z$  and  $f_z$ ", these three sets of "real data vs fake data" must be cooperatively used. It is notable that there

is no activation function in the last layer of the discriminator since WGAN-GP needs to realize an approximate fitting of the Wasserstein distance.

## 2.5 The procedure of SAGAN

The procedure of SAGAN is displayed in Fig. 4.



**Fig. 4.** The procedure of SAGAN.

Step 1: Initialize the SAGAN model.

Step 2: Preprocess the input TI of 3D stochastic media to create three-direction 2D slice-image datasets.

Step 3: Feed Gaussian noise into  $G$  to generate fake 3D images.

Step 4: Train  $D$  with real 2D images sliced from real 3D images, and fake 2D images sliced from fake 3D images, respectively.

Step 5: If the current epoch dose not surpass the predetermined maximum epoch number, go to

step 3; otherwise, go to step 6.

Step 6: Save the SAGAN model, and use it to generate reconstructed images.

### 3. Evaluation criteria of reconstruction quality

To evaluate reconstruction quality, the statistical and morphological characteristics of material microstructures are compared between reconstructions and the original TI [24]. Meanwhile, it is also necessary to compare with other methods. Since SAGAN extracts features from only one TI, it is reasonable to compare with typical one-TI-based reconstruction methods. Hence, ConSinGAN [20], single normal equation simulation (SNESIM) [25], and image quilting (IQ) [26] are chosen to achieve stochastic reconstructions to confirm SAGAN's reconstruction effectiveness. All the methods mentioned above are trained using a single image as the TI.

We use multiple-point connectivity (MPC) [27] to assess the connectivity between multiple points, which is defined as:

$$MPC(N) = E\{I(x) \cdot I(x + d) \cdots I(x + (N - 1) * d)\} = E\{\prod_{j=0}^{N-1} I(x + j * d)\}, \quad (5)$$

where  $N$  represents the number of voxels in one certain direction;  $x$  represents the position of one specific voxel in space;  $d$  is the “lag distance” between neighboring voxels;  $I(x)$  is the attribute value. For a two-phase condition, like pores and grains, if  $x$  is located in pores,  $I(x) = 1$ ; otherwise  $I(x) = 0$ . The  $n$ -point correlation function [28] is very similar to the definition of MPC. MPC is actually one of the particular cases of the “ $n$ -point correlation function” when  $n = N$ . Compared to the first several orders of correlation functions, we use MPC to estimate the correlation of structures from a larger and broader perspective since  $N$  is relatively larger.

Frechet inception distance (FID) [29] is commonly employed to assess the quality of the reconstructed image and measured by the differences between real and generated datasets, defined as:

$$FID = \|\mu_r - \mu_g\|_2^2 + Tr(\Sigma r + \Sigma g - 2\sqrt{\Sigma r \Sigma g}), \quad (6)$$

where  $\mu_r$  and  $\Sigma r$  denote the mean and covariance matrix of the feature vectors of the real image  $r$  output by the Inception Net-V3 network [30];  $\mu_g$  and  $\Sigma g$  denote the mean and covariance matrix of the generated image  $g$  output also by the Inception Net-V3 network, and  $Tr(\cdot)$  denotes the sum of the diagonal elements of the matrix.

The process of calculating FID is as follows: 1. Sample randomly from the real data distribution and the generative distribution, respectively. 2. Use the pre-trained Inception Net-V3 network to extract feature vectors from these samples. 3. The extracted feature vectors are calculated in accordance with Eq. (6) to obtain the final FID scores.

Pores have complex and highly irregular geometrical profiles. The shape factor  $G$  [31] is used to define its shape:

$$G = \frac{VL}{A_s^2}, \quad (7)$$

where  $A_s$  represents the surface area of the pore;  $V$  is the pore volume and  $L$  represents the length between two farthest points within it.

Other parameters such as porosity, tortuosity, and Minkowski functionals are also studied in the following tests. Porosity and tortuosity are calculated using the software Avizo [32]. Minkowski functionals are calculated using MorphoLibJ [33], an open-source image processing library of ImageJ.

Diversity is another crucial metric for assessing the quality of generated images since one-TI-based methods are possibly prone to overfitting results. Building upon multi-scale sliced Wasserstein distance (MS-SWD) and multi-dimensional scaling (MDS), Song et al [34, 35] fused MS-SWD with MDS, called MS-SWD-MDS, to project reconstruction results into 2D space for convenient visual assessment of diversity. We use MS-SWD-MDS to illustrate the distributions of the TI and the reconstructed images by projecting them into a 2D space, enabling visualization of the disparity or similarity between them.

## 4. Results and discussion

### 4.1 Training Data and parameter setting

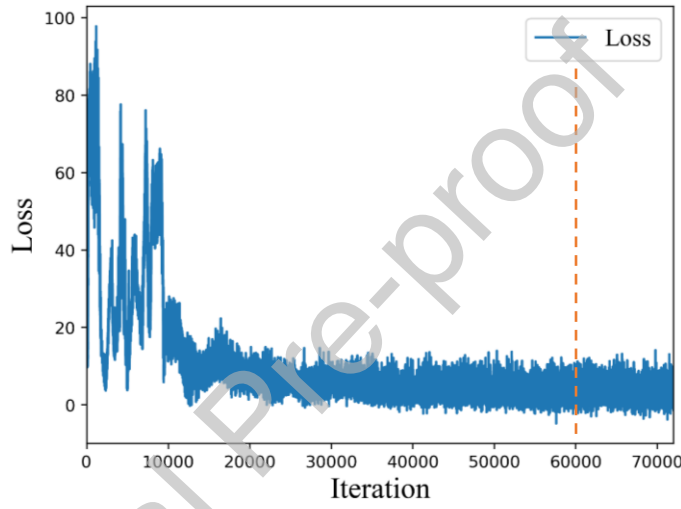
We use two real 3D datasets of stochastic microstructures as TIs, one of which is the public benchmark dataset named Berea sandstone (resolution: 3  $\mu\text{m}$  per voxel), provided by the Department of Earth Science and Engineering at Imperial College London [10]. The other is from real industrial composite material (resolution: 13.4  $\mu\text{m}$  per voxel). The first one represents open-cell continuous porous media and is used to mimic the internal morphology of porous media, called Berea\_128 (size:  $128^3$  voxels) hereafter. The second one represents closed-cell porous structures with packed spherical pores, called Composite\_128 (size:  $128^3$  voxels) hereafter.

To provide enough TIs for training but reducing the hardware burdens, we did not directly use 128 2D images ( $128 \times 128$  pixels each) along the X, Y, and Z directions as TIs. In the real experimental process, for example, in the X direction, we first randomly select a 2D image ( $128 \times 128$ ) along the X direction; then randomly cut a  $64 \times 64$  portion from this one. We repeat this process for 4800 times to extract the 64-sized 2D image dataset along the X direction. Then, we perform the similar process respectively along the Y and Z directions. Finally, we attain three 64-sized 2D image datasets along the X, Y and Z directions, totally with 14400 2D 64-sized images, which are used as the TIs for training.

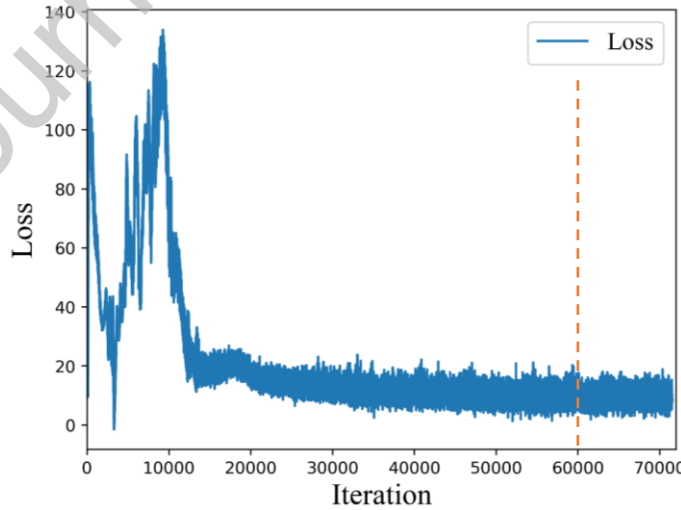
In this way, on the one hand, enough 2D images can be extracted as TIs, and on the other hand, the relatively smaller 64-sized images will not greatly challenge the computational burdens. The proper randomness in selecting 64-sized sub-images in the original 2D 128-sized image can

bring proper diversity to the reconstructed results. Different from the way of SAGAN's attaining its 64-sized 2D TI datasets, in the following experiments, the other three methods (SNESIM, IQ, and ConSinGAN) used for comparison all use the whole Berea\_128 or Composite\_128 as the single TI. The reconstructions by SNESIM, IQ, ConSinGAN and SAGAN have the size of  $128^3$  and will be compared with Berea\_128 or Composite\_128 for quality estimation.

Fig. 5 shows the loss (calculated through the loss function Eq. (4)) in training for Berea\_128 and Composite\_128, respectively. It seems that the loss tends to level off between 60000 and 70000 iterations, so we choose 60000 as the total number of iterations. Note that an epoch may consists of many iterations, and in our experiments each epoch iterates 600 times, thus determining the maximum epoch to be 100 ( $= 60000/600$ ).



(a) Berea\_128



(b) Composite\_128

**Fig. 5.** The loss of Berea\_128 and Composite\_128.

In most scenarios, the parameters of networks are validated based on practical experiments or

empirical values. The hardware equipment includes an NVIDIA RTX GPU 3080 (10 GB graphic memory), an Intel Core i5-8250U CPU (1.60 GHz), and 16 GB RAM. The software platform is PyTorch 1.7.0.

#### 4.2 Reconstruction of open-cell stochastic microstructure

In this section, Berea\_128 is used for open-cell stochastic experiments. To attain experimental statistics from an average angle, for each comparative methods and SAGAN, we generate ten results based on Berea\_128.

##### 4.2.1 Comparison of FID

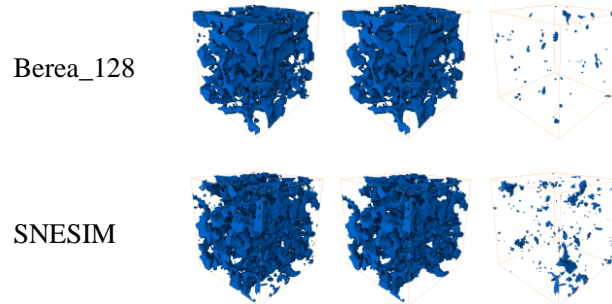
FID is a metric utilized to gauge the resemblance between two distributions of images, thereby employed to assess the quality of the reconstructed images. A lower FID value suggests that the distribution of the generated image is more similar to that of the original TI. Table 1 depicts the mean FIDs of ten reconstructions by different four methods. SAGAN has the lowest FID, which indicates that the quality of SAGAN's reconstructions is best. Note that the best statistics are highlighted with bold numbers in the tables of this paper.

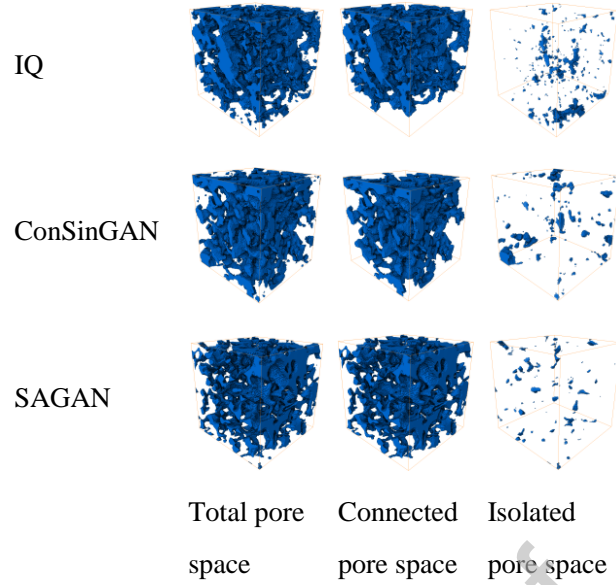
**Table 1** Average FIDs of ten reconstructions by different four methods.

	SNESIM	IQ	ConSinGAN	SAGAN
FID	153.89	138.37	136.56	<b>97.38</b>

##### 4.2.2 Comparison of pore space

The pore space (i.e., total pore space, connected pore space, and isolated pore space) of Berea\_128 and one reconstruction of the four methods are extracted using Avizo [32], as shown in Fig. 6. Note that the total pore space equals the sum of connected pore space and isolated pore space. Table 2 demonstrates the mean porosities of total pore space, connected pore space, and isolated pore space of Berea\_128 and ten reconstructions by the four different methods, respectively. Overall, it can be visualized that the SAGAN's reconstructions are closest to Berea\_128.





**Fig. 6.** Total pore space, connected pore space, and isolated pore space of Berea\_128 and one reconstruction by SNESIM, IQ, ConSinGAN, and SAGAN.

**Table 2** Mean porosities of total pore space, connected pore space, and isolated pore space of Berea\_128 and ten reconstructions by SNESIM, IQ, ConSinGAN, and SAGAN.

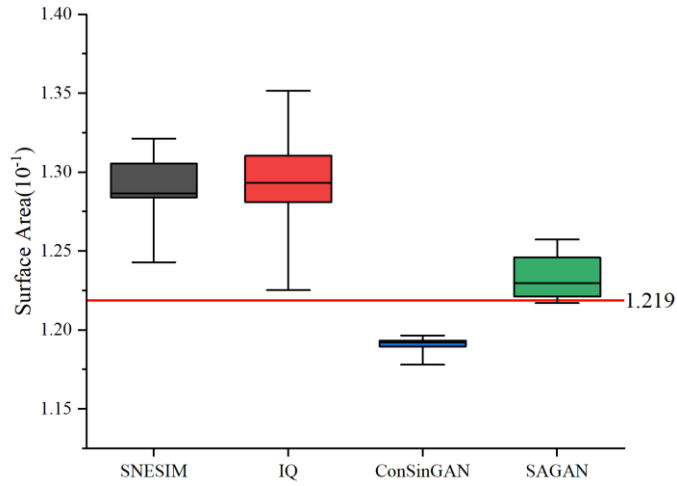
	Berea_128	SNESIM	IQ	ConSinGAN	SAGAN
Total pore space (%)	20.69	21.61	21.99	20.49	<b>20.55</b>
Connected pore space (%)	20.58	21.03	21.39	19.74	<b>20.32</b>
Isolated pore space (%)	0.11	0.58	0.60	0.75	<b>0.23</b>

#### 4.2.3 Comparison of morphology

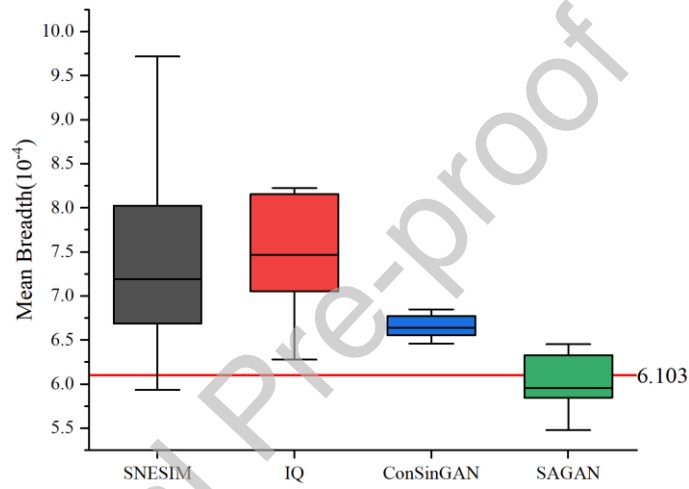
Minkowski functionals, employed as morphological descriptors, can present both local and global morphological details of stochastic microstructures. The 3D topology of porous microstructure is effectively characterized through the calculation of three Minkowski functionals: surface area, mean breadth, and Euler characteristic [14]. Surface area, normalized against total volume, is instrumental in capturing the overall geometric features. Mean breadth serves as an indicator of resistance to flow within the tubular network, and Euler characteristic can depict the connectivity of porous microstructure. The reference [10] described the computation of the three Minkowski functionals in detail.

Fig. 7 displays boxplots illustrating the three Minkowski functionals of Berea\_128 and ten reconstructions generated by SNESIM, IQ, ConSinGAN, and SAGAN. The Berea\_128's three Minkowski functionals are displayed by red lines. It is seen that the reconstructions of SAGAN bear the closest resemblance to Berea\_128.

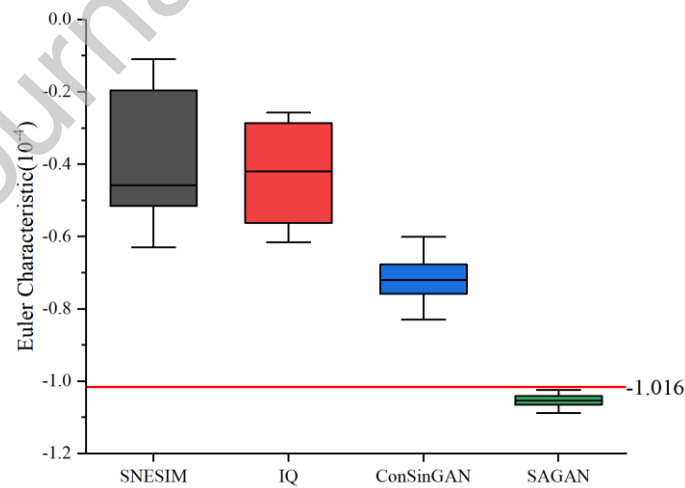




(a) surface area



(b) mean breadth



(c) Euler characteristic

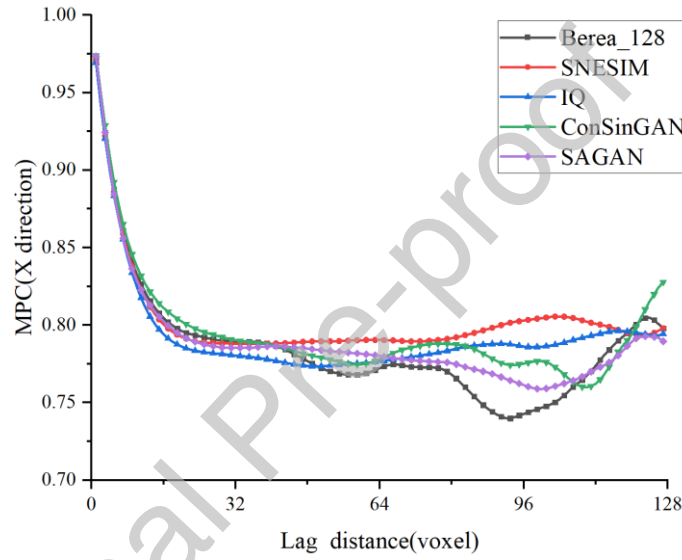
**Fig. 7.** The boxplots depicting the three Minkowski functionals of Berea\_128 and reconstructions of SNESIM, IQ, ConSinGAN, and SAGAN. The Berea\_128's three Minkowski functionals are displayed by red lines.

#### 4.2.4 Comparison of MPC

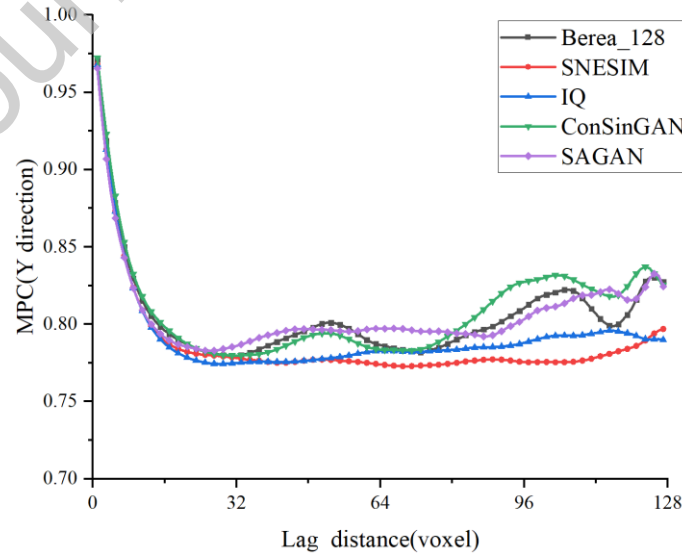
Fig. 8 illustrates the average MPC curves of Berea\_128 and each reconstruction method in the X, Y, and Z directions. To visualize which method is closest to Berea\_128, a difference degree ( $DD$ ) is defined as follows:

$$DD(TI, ri) = \sum_{j=1}^n (X_j - x_j)^2, \quad (8)$$

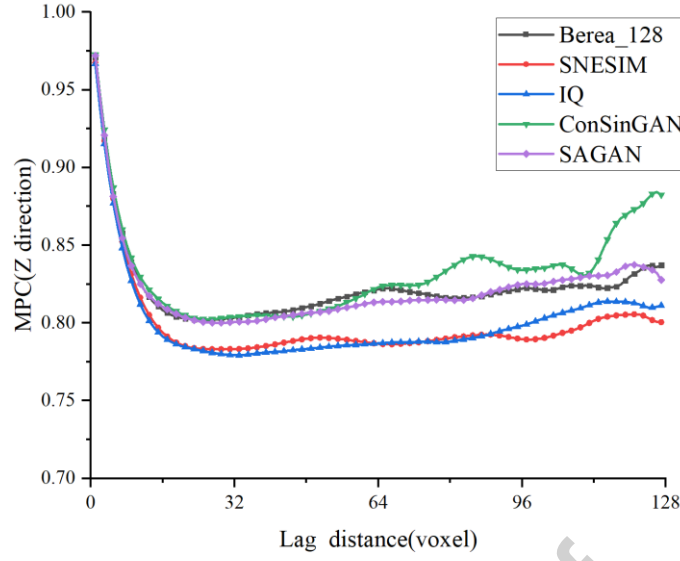
where  $TI$  denotes the training image and  $ri$  represents the generation result of the four methods;  $X_j$  and  $x_j$  denote the MPC values of the  $TI$  and generated images with a distance  $j$  (i.e., lag distance), respectively. According to the MPC curves in Fig. 8, the  $DD$ s between Berea\_128 and reconstructions are computed and shown in Table 3, indicating that the reconstruction results of SAGAN have the smallest error to Berea\_128 in all three directions.



(a) X direction



(b) Y direction



(c) Z direction

**Fig. 8.** MPC curves of Berea\_128 and average MPC curves of each reconstruction method in X, Y, and Z directions.

**Table 3** *DDs* of the MPC curves (shown in Fig. 8) of Berea\_128 and the reconstruction results by four methods.

Direction	<i>DD</i>			
	SNESIM	IQ	ConSinGAN	SAGAN
X	0.0937	0.0498	0.0296	<b>0.0128</b>
Y	0.0605	0.0328	0.0097	<b>0.0084</b>
Z	0.0780	0.0713	0.0352	<b>0.0027</b>

#### 4.2.5 Comparison of tortuosity

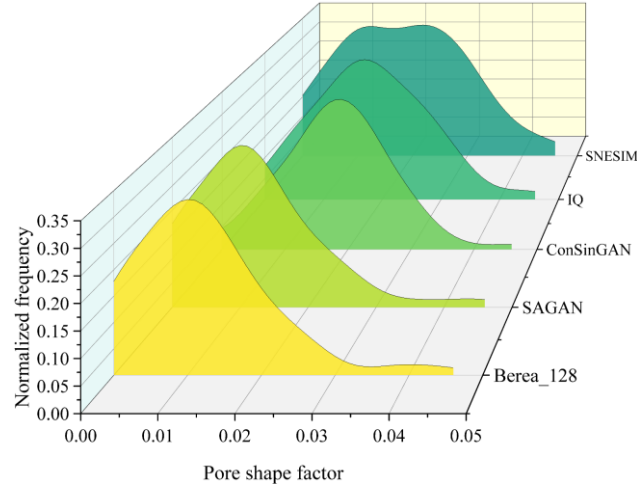
In the stochastic media, tortuosity is a property of characterizing the tortuous degree of pore space. Table 4 shows the average tortuosity values for Berea\_128 and the reconstructions of each method. Clearly, the tortuosity of SAGAN is closest to Berea\_128.

**Table 4** The average tortuosity values for Berea\_128 and the reconstructions of each method.

	Berea_128	SNESIM	IQ	ConSinGAN	SAGAN
Tortuosity	1.584	1.641	1.628	1.606	<b>1.592</b>

#### 4.2.6 Comparison of pore shape factors

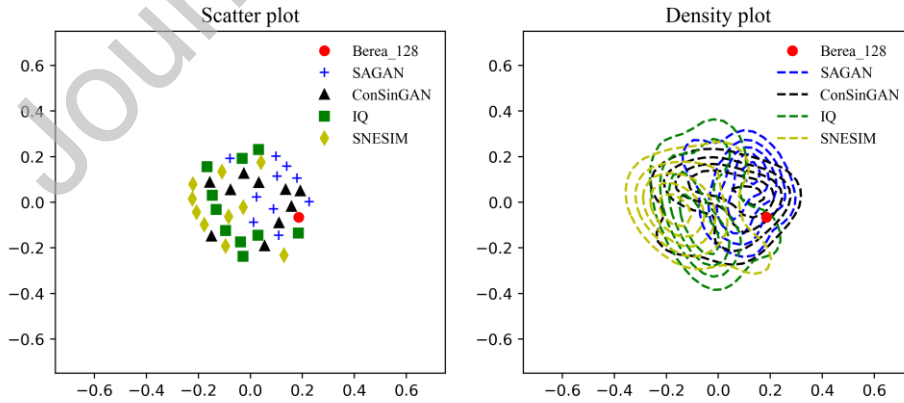
Real pores have complex and high irregular geometrical profiles. We use a dimensionless shape factor to describe their approximate shapes. Fig. 9 shows the average distributions of pore shape factors for Berea\_128 and the reconstructions of each method. It can be visualized that the shape factor distribution of SAGAN is closest to that of Berea\_128, indicating that its reconstruction is the best.



**Fig. 9.** The average distributions of pore shape factors for Berea\_128 and the reconstructions of each method.

#### 4.2.7 Evaluation of diversity

MS-SWD-MDS is used to project Berea\_128 and reconstruction results into 2D space for easy visual observation of diversity using scatter plots and density plots. Each point in the scatter plots signifies a 3D image of Berea\_128 or reconstructions. Each contour line in the density plots shows the distribution of Berea\_128 or reconstructions. Fig. 10 depicts the scatter plots and density plots of Berea\_128 and reconstruction results, respectively. It can be visualized that the distributions of SAGAN are scattered but closest to Berea\_128. On the one hand, the degree of dispersion proves the high diversity of reconstruction results, thus avoiding the risk of overfitting. On the other hand, the closeness to Berea\_128 also proves the reasonable resemblance between Berea\_128 and reconstruction of SAGAN.

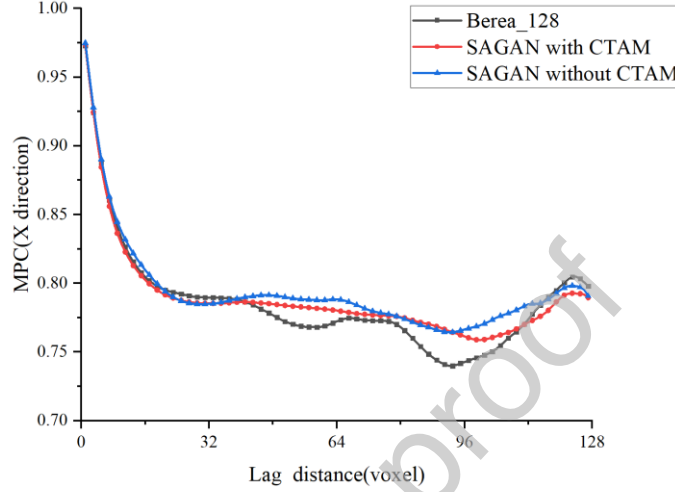


**Fig. 10.** The scatter plots and density plots of Berea\_128 and reconstruction results of SNESIM, IQ, ConSinGAN, and SAGAN.

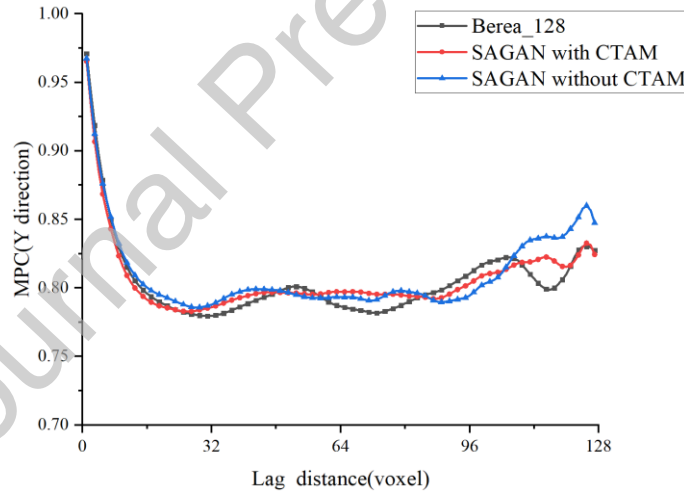
#### 4.2.8 Ablation experiments of CTAM based on Berea\_128

To evaluate the functionality of CTAM, we train two SAGAN models: one model with

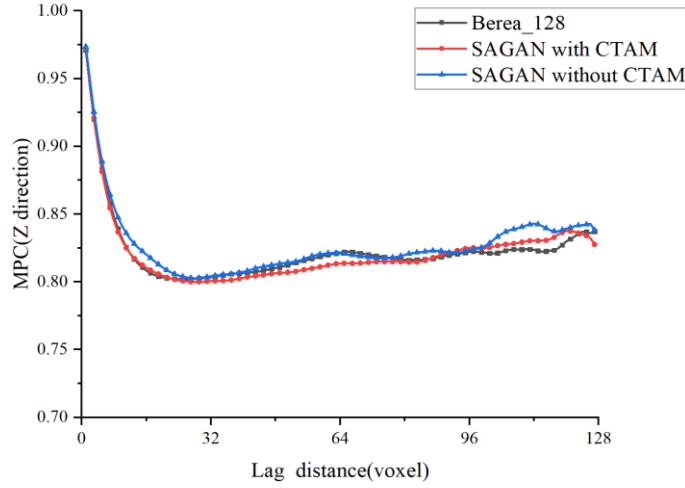
CTAM and one without CTAM. For each model, 10 reconstructions are conducted using Berea\_128 to obtain mean statistics. Fig. 11 shows the average MPC curves of Berea\_128 and two SAGAN models (with and without CTAM, respectively). Table 5 displays the *DDs* of MPC curves (shown in Fig. 11) between the reconstructions and Berea\_128 in the X, Y, and Z directions. It appears that “SAGAN with CTAM” is better than “SAGAN without CTAM” in all three directions, indicating the effectiveness of utilizing CTAM for the reconstruction.



(a) X direction



(b) Y direction



(c) Z direction

**Fig. 11.** Average MPC curves of Berea\_128 and two SAGAN models (with and without CTAM, respectively).

**Table 5** *DDs* of the MPC curves (shown in Fig. 11) of Berea\_128 and the reconstruction results by two SAGAN models.

Direction	<i>DD</i>	
	SAGAN with CTAM	SAGAN without CTAM
X	<b>0.0128</b>	0.0213
Y	<b>0.0084</b>	0.0236
Z	<b>0.0027</b>	0.0069

#### 4.2.9 Comparison of CPU/GPU/memory usage and running time

As shown in Table 6, the information on the GPU/CPU usage, memory usage, GPU graphic memory and the reconstruction time for ten reconstructions by each method are presented. Although SAGAN and ConSinGAN obviously spend more time in the first reconstruction compared to traditional methods SNESIM and IQ, their subsequent nine reconstructions is much faster thanks to the reuse of their established model and parameters in the first training. Unfortunately, the traditional methods SNESIM and IQ are incapable of reusing their parameters established in the first training, so their multiple reconstruction tasks will be quite ineffective once the quantity is huge. Note that thanks to the 2D training datasets of SAGAN, the maximal GPU graphic memory by SAGAN is much less than ConSinGAN.

**Table 6** The reconstruction time, GPU graphic memory, and CPU/GPU usage for each method based on Berea\_128 in the reconstruction of ten results.

	SNESIM	IQ	ConSinGAN	SAGAN
Maximal CPU usage	64%	70%	<b>37%</b>	<b>37%</b>
Maximal GPU usage	<b>0</b>	<b>0</b>	99%	99%

Maximal memory usage	60%	65%	<b>40%</b>	<b>40%</b>
Maximal GPU graphic memory (MB)	<b>0</b>	<b>0</b>	9456	5647
Time taken for the initial reconstruction (s)	<b>1530</b>	1829	25635	22162
Time for the other 9 reconstructions (s)	12563	15632	4	<b>2</b>

### 4.3 Reconstruction of closed-cell stochastic microstructure

Similar to Section 4.2, in this section, Composite\_128 is used for closed-cell stochastic experiments. To attain average experimental statistics, we generate ten results based on Composite\_128 for each comparative methods and SAGAN.

#### 4.3.1 Comparison of FID

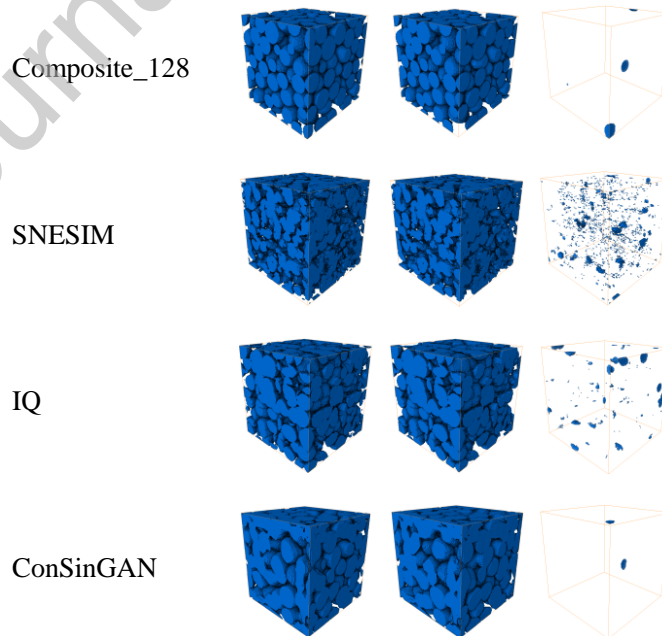
FID is used to estimate the quality of the reconstructed images. As illustrated in Table 7, SAGAN has the lowest FID, which indicates the best quality of the generated image.

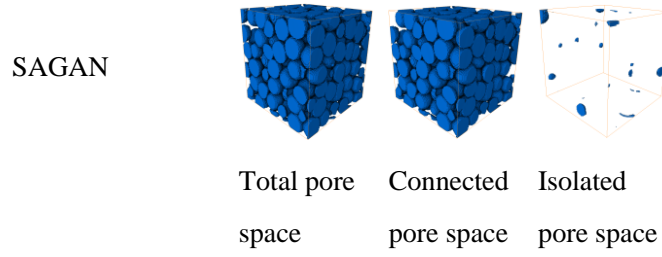
**Table 7** Average FIDs of ten reconstructions by different four methods.

	SNESIM	IQ	ConSinGAN	SAGAN
FID	568.86	535.23	204.18	<b>124.48</b>

#### 4.3.2 Comparison of pore space

The total pore space, connected pore space, and isolated pore space of Composite\_128 and one reconstruction of the four methods are extracted by Avizo [32], as shown in Fig. 12. Table 8 is the average porosities of total pore space, connected pore space, and isolated pore space for Composite\_128 and different reconstruction methods, respectively. It can be visualized that SAGAN is closest to Composite\_128 in the “total pore space” and “connected pore space”.





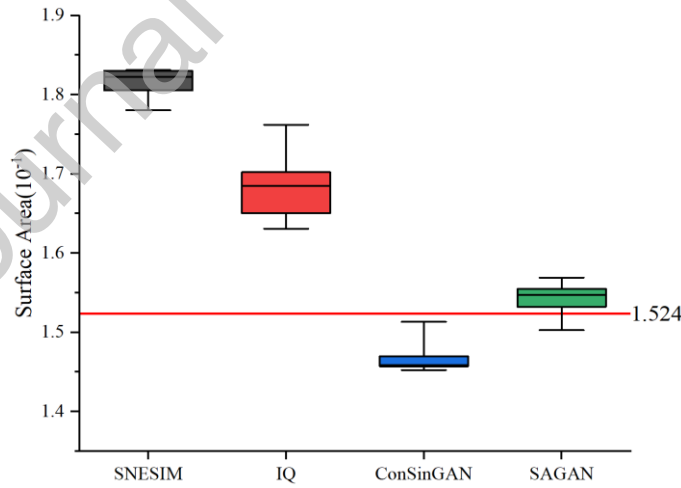
**Fig. 12.** Total pore space, connected pore space, and isolated pore space of Composite\_128 and one reconstruction of SNESIM, IQ, ConSinGAN and SAGAN.

**Table 8** Mean porosities of total pore space, connected pore space, and isolated pore space of Composite\_128 and ten reconstructions by SNESIM, IQ, ConSinGAN, and SAGAN.

	Composite_128	SNESIM	IQ	ConSinGAN	SAGAN
Total pore space (%)	61.33	50.43	60.16	62.06	<b>60.72</b>
Connected pore space (%)	61.26	50.14	60.04	61.96	<b>60.57</b>
Isolated pore space (%)	0.07	0.29	0.12	<b>0.10</b>	0.15

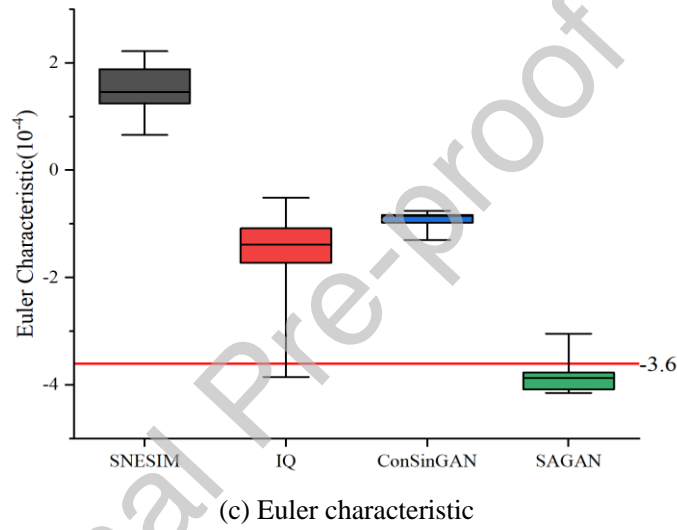
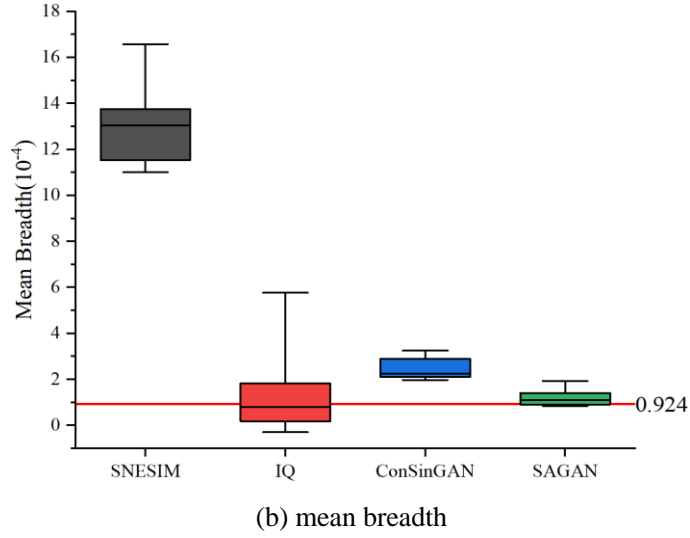
#### 4.3.3 Comparison of morphology

Minkowski functionals are utilized to characterize the morphological information of Composite\_128. The Minkowski functionals of Composite\_128 and the reconstruction results of SNESIM, IQ, ConSinGAN, and SAGAN are illustrated in Fig. 13. The Composite\_128's three Minkowski functionals are displayed by red lines. The values and error ranges are most similar between Composite\_128 and the SAGAN reconstructions.



(a) surface area

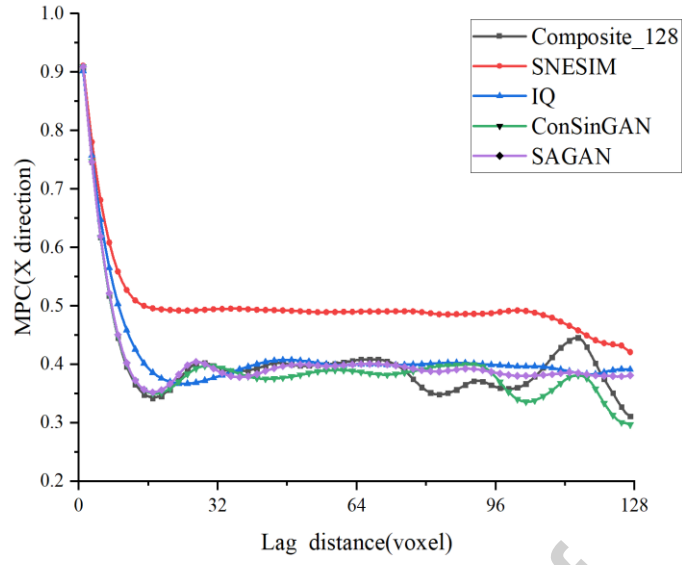




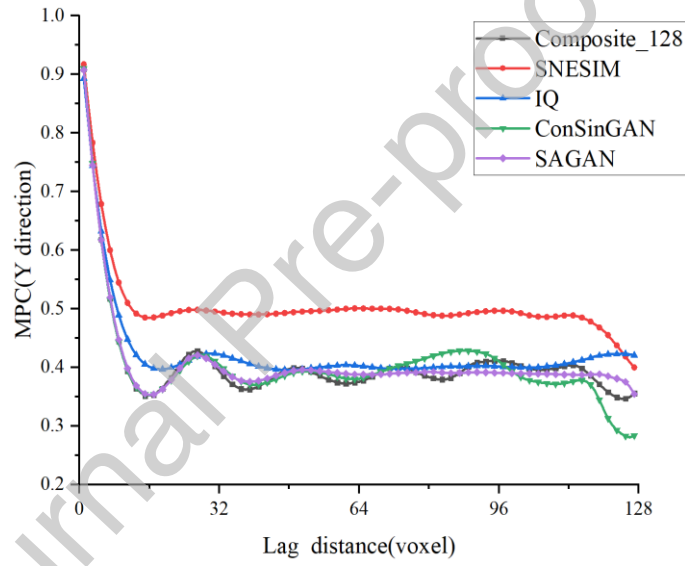
**Fig. 13.** The boxplots depicting the three Minkowski functionals of Composite\_128 and reconstructions of SNESIM, IQ, ConSinGAN, and SAGAN. The Composite\_128's three Minkowski functionals are displayed by red lines.

#### 4.3.4 Comparison of MPC

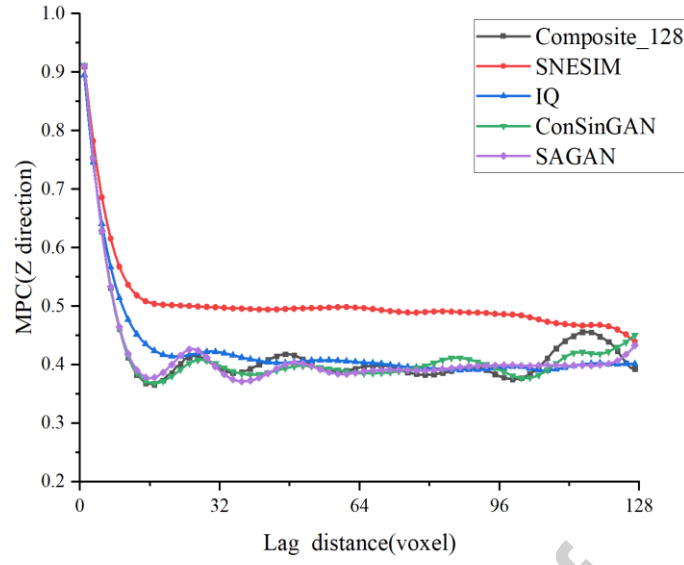
Fig. 14 illustrates the average MPC curves of Composite\_128 and each reconstruction method in the X, Y, and Z directions. The  $DDs$  are calculated based on the MPC curves in Fig. 14 and illustrated in Table 9. It can be observed that reconstruction results of SAGAN have the smallest error to Composite\_128 (best in the X and Y directions).



(a) X direction



(b) Y direction



(c) Z direction

**Fig. 14.** MPC curves of Composite\_128 and average MPC curves of each reconstruction method in the three directions.

**Table 9** *DDs* of the MPC curves (shown in Fig. 14) of Composite\_128 and the reconstruction results by four methods.

Direction	<i>DD</i>			
	SNESIM	IQ	ConSinGAN	SAGAN
X	1.3569	0.1350	0.0969	<b>0.0633</b>
Y	1.2889	0.1082	0.0595	<b>0.0170</b>
Z	1.0960	0.0947	<b>0.0330</b>	0.0471

#### 4.3.5 Comparison of tortuosity

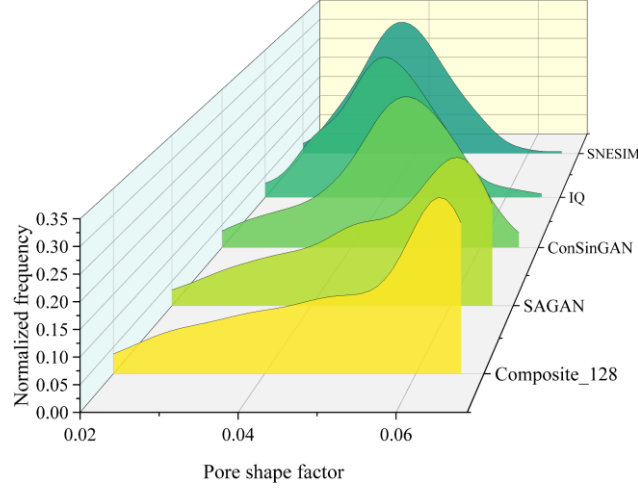
The average tortuosity values for Composite\_128 and the reconstructions of each method are shown in Table 10, in which the tortuosity of SAGAN is closest to Composite\_128.

**Table 10** The average tortuosity values for Composite\_128 and the reconstructions of each method.

	Composite_128	SNESIM	IQ	ConSinGAN	SAGAN
Tortuosity	1.559	1.595	1.587	1.525	<b>1.536</b>

#### 4.3.6 Comparison of pore shape factors

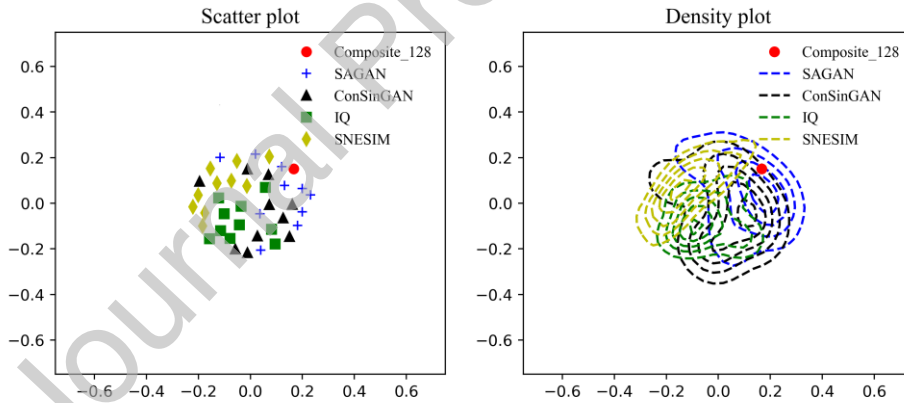
Fig. 15 shows the average distributions of pore shape factors for Composite\_128 and the reconstructions of each method. It can be visualized that the shape factor distribution of SAGAN is closest to Composite\_128.



**Fig. 15.** The average distributions of pore shape factors for Composite\_128 and the reconstructions of each method.

#### 4.3.7 Evaluation of diversity

Fig. 16 shows the scatter plots and density plots of Composite\_128 and reconstruction results, respectively. It can be visualized that the distributions of SAGAN are diverse enough but closest to Composite\_128, displaying a preferable balance between diversity and resemblance than the other three methods.

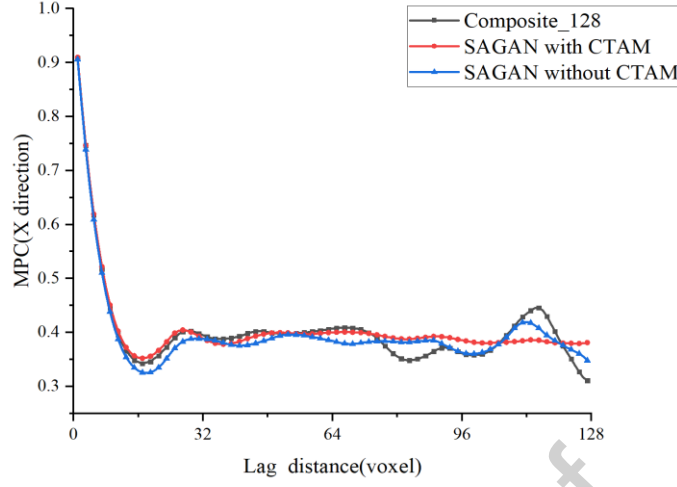


**Fig. 16.** The scatter plots and density plots of Composite\_128 and reconstruction results of SNESIM, IQ, ConSinGAN, and SAGAN.

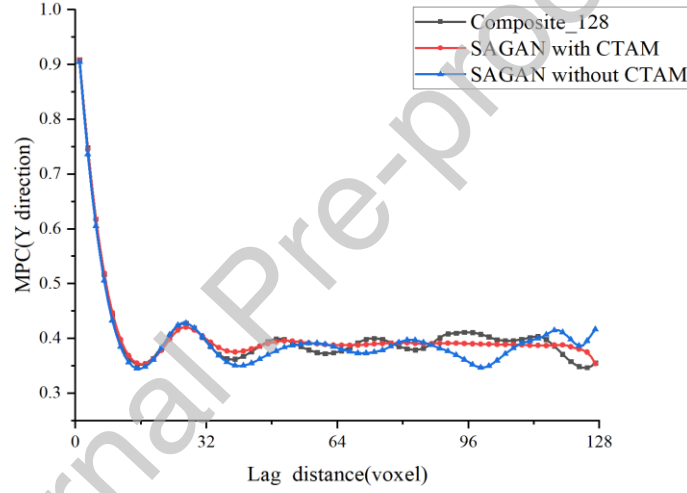
#### 4.3.8 Ablation experiments of CTAM based on Composite\_128

To test the function of CTAM using Composite\_128, we still train two models of SAGAN: one model is with CTAM, while the other is trained without CTAM. For each model, 10 reconstructions are conducted using Composite\_128 to obtain mean statistics. Fig. 17 illustrates the MPC curves of Composite\_128 and two SAGAN models (with CTAM and without CTAM respectively). Table 11 illustrates the *DDs* of MPC curves (shown in Fig. 17) between the reconstructions and Composite\_128. It seems that “SAGAN with CTAM” is better than “SAGAN

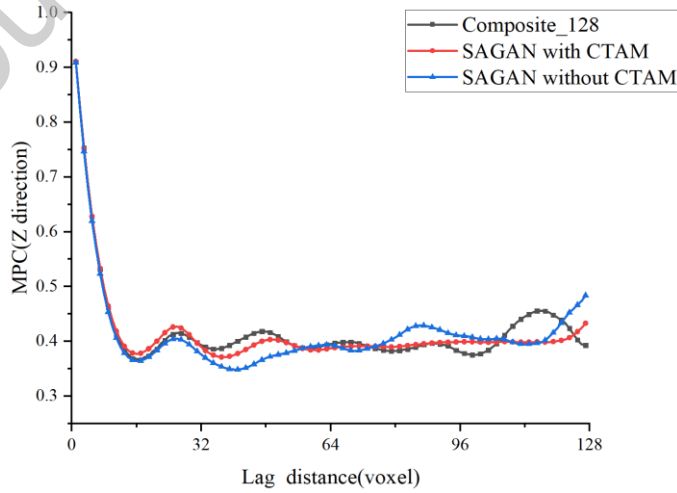
without CTAM” in two directions (Y and Z), but they do not have much differences in the X direction, indicating the overall better effectiveness of utilizing CTAM for the reconstruction.



(a) X direction



(b) Y direction



(c) Z direction

**Fig. 17.** Average MPC curves of Composite\_128 and two SAGAN models (with and without

CTAM, respectively).

**Table 11** *DDs* of the MPC curves (shown in Fig. 17) of Composite\_128 and the reconstruction results by two SAGAN models.

Direction	<i>DD</i>	
	SAGAN with CTAM	SAGAN without CTAM
X	0.063	<b>0.041</b>
Y	<b>0.017</b>	0.071
Z	<b>0.047</b>	0.113

#### 4.3.9 Comparison of CPU/GPU/memory usage and running time

As shown in Table 12, the information on the GPU/CPU usage, memory usage, and reconstruction time based on Composite\_128 for each method is presented. The deep learning methods show evident advantages in multiple reconstructions once the first training is finished. Still, thanks to the 2D training datasets of SAGAN, the maximal GPU graphic memory by SAGAN is much less than ConSinGAN.

**Table 12** The reconstruction time, GPU graphic memory, and CPU/GPU usage based on Composite\_128 for each method in the reconstruction of ten results.

	SNESIM	IQ	ConSinGAN	SAGAN
Maximal CPU usage	63%	70%	<b>36%</b>	<b>36%</b>
Maximal GPU usage	<b>0</b>	<b>0</b>	99%	99%
Maximal memory usage	59%	65%	41%	<b>40%</b>
Maximal GPU graphic memory (MB)	<b>0</b>	<b>0</b>	9765	5749
Time taken for the initial reconstruction (s)	<b>2213</b>	2513	27928	23285
Time for the other 9 reconstructions (s)	19836	22683	4	<b>2</b>

#### 4.4 Comparison of computational burdens with SAGAN using 3D TIs

In sections 4.2 and 4.3, both as deep learning methods, ConSinGAN and SAGAN are compared to evaluate their reconstruction quality. ConSinGAN relies on only one 3D image as the TI, using multiple stages to reconstruct images, so it spends much more time for reconstruction due to its multiple stages. It seems that it is unfair to compare SAGAN and ConSinGAN in the time and memory performance since ConSinGAN performs normally much longer. Therefore, we have added the comparative experiments with SAGAN directly using 3D images as TIs, called “SAGAN\_3D” for convenience.

Now we mainly focus on the time and memory (both computer memory and GPU graphic memory) performance of SAGAN and SAGAN\_3D. For comparison, we use the same TI datasets (Berea\_128 and Composite\_128) to reconstruct ten results ( $128^3$  each). Note that, for SAGAN,

there are totally 14400 2D 64-sized images randomly selected from the 128-cubed Berea\_128 and Composite\_128, respectively. For SAGAN\_3D, we randomly select 225 ( $=14400/64$ ) 3D 64-cubed images from Berea\_128 and Composite\_128, respectively. Hence, the TIs for SAGAN and SAGAN\_3D are approximately equivalent. In this situation, the main difference between SAGAN\_3D and SAGAN lies in that SAGAN\_3D uses the whole 3D 64-cubed images as TIs instead of slicing them into 2D images. On the contrary, SAGAN slices 3D images into 2D images as TIs.

As shown in Table 13, the information on the GPU/CPU usage, memory usage, GPU graphic memory and the reconstruction time for ten reconstructions by SAGAN and SAGAN\_3D are presented. Note that thanks to the 2D training datasets of SAGAN, the maximal GPU graphic memory by SAGAN is much less than SAGAN\_3D, but SAGAN does not show evident speed advantage over SAGAN\_3D.

**Table 13** The reconstruction time, GPU graphic memory, and CPU/GPU usage based on Berea\_128 and Composite\_128 by SAGAN and SAGAN\_3D in the reconstruction of ten results.

	Berea_128		Composite_128	
	SAGAN	SAGAN_3D	SAGAN	SAGAN_3D
Maximal CPU usage	<b>37%</b>	38%	<b>36%</b>	37%
Maximal GPU usage	99%	99%	99%	99%
Maximal memory usage	<b>40%</b>	43%	<b>40%</b>	43%
Maximal GPU graphic memory (MB)	<b>5647</b>	7709	<b>5749</b>	7831
Time taken for the initial reconstruction (s)	<b>22162</b>	22957	<b>23285</b>	24036
Time for the other 9 reconstructions (s)	<b>2</b>	3	<b>2</b>	3

## 5. Conclusion and future work

In this paper, a model SAGAN for the 3D reconstruction of stochastic microstructures is developed and successfully tested for the case studies of open-cell and closed-cell stochastic structures. The whole model needs only a 3D image as the TI and slice it into three 2D slice datasets along the X, Y and Z directions, by which the maximal GPU graphic memory is obviously reduced and the quantity of training data largely decreases. CTAM captures cross-dimensional interactions to prioritize the learned features and improves the effectiveness of training. The experiments show that the reconstructed images of SAGAN retain the characteristics of the original training data. Due to the 2D training datasets of SAGAN, the maximal GPU graphic memory by SAGAN is much less than ConSinGAN. SAGAN also show advantages in training time for multiple reconstructions over traditional reconstruction methods once the first training is finished. Hence, the outcomes validate the applicability of the proposed method for the 3D reconstruction of stochastic microstructures.

There are two directions for our future work. First, the tuning of hyperparameters is a big challenge, so using some empirical parameters or models are highly recommended. We will further focus on the relation between initial parameters and final ones, hopefully establishing reliable principles to build proper empirical parameters or models. Secondly, since we use three different training datasets, we may use three discriminators to capture the different features of the three training datasets respectively instead of using only one discriminator, which surely will increase the hardware load of the model and training time. To address this issue, we will conduct further studies to find a proper balance between reconstruction effect and computational burdens of three discriminators.

#### **Authorship Contribution Statement**

**Ting Zhang:** Conceptualization, Methodology, Writing–original draft. **Ningjie Bian:** Methodology, Writing–original draft. **Xue Li:** Investigation, Writing – review & editing.

#### **Declaration of Competing Interest**

The authors declare that they have no conflicts of interest.

#### **Acknowledgments**

This work is supported by the National Natural Science Foundation of China (Nos. 41672114, 41702148, 22108269).

#### **Data Availability Statement**

The data used to support this study are available from the corresponding author upon request.

#### **References**

- [1] Lin W, Li X, Yang Z, et al. Construction of dual pore 3-D digital cores with a hybrid method combined with physical experiment method and numerical reconstruction method. *Transport in porous media* 2017; 120: 227-238.
- [2] Lymberopoulos DP, Payatakes AC. Derivation of topological, geometrical, and correlational properties of porous media from pore-chart analysis of serial section data. *Journal of colloid and interface science* 1992; 150(1): 61-80.
- [3] Fredrich JT, Lindquist WB. Statistical characterization of the three-dimensional microgeometry of porous media and correlation with macroscopic transport properties. *International journal of rock mechanics and mining sciences* 1997; 34(3-4).
- [4] Hou J, Zhang SK, Sun RY, et al. Reconstruction of 3D network model through CT scanning SPE Europe featured at EAGE Conference and Exhibition 2007: SPE-106603-MS.
- [5] Hazlett RD. Statistical characterization and stochastic modeling of pore networks in relation to fluid flow. *Mathematical geology* 1997; 29: 801-822.
- [6] Wu K, Nunan N, Crawford JW, et al. An efficient Markov chain model for the simulation of heterogeneous soil structure. *Soil Science Society of America Journal* 2004; 68(2): 346-351.



- [7] Okabe H, Blunt MJ. Pore space reconstruction using multiple-point statistics. *Journal of petroleum science and engineering* 2005; 46(1-2): 121-137.
- [8] Feng J, Teng Q, He X, et al. Accelerating multi-point statistics reconstruction method for porous media via deep learning. *Acta Materialia* 2018; 159: 296-308.
- [9] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial networks. *Communications of the ACM* 2020; 63(11): 139-144. <http://dx.doi.org/10.48550/arxiv.1406.2661>.
- [10] Mosser L, Dubrulle O, Blunt MJ. Reconstruction of three-dimensional porous media using generative adversarial neural networks, *Berea Data. Physical Review E* 2017; 96(4): 043309.
- [11] Volkhonskiy D, Muravleva E, Sudakov O, et al. Reconstruction of 3d porous media from 2d slices. 2019; arXiv preprint arXiv:1901.10233.
- [12] Shams R, Masihi M, Boozarjomehry RB, et al. Coupled generative adversarial and auto-encoder neural networks to reconstruct three-dimensional multi-scale porous media. *Journal of Petroleum Science and Engineering* 2020; 186: 106794. <https://doi.org/10.1016/j.petrol.2019.106794>.
- [13] Kononov E, Tashkinov M, Silberschmidt VV. Reconstruction of 3D Random Media from 2D Images: Generative Adversarial Learning Approach. *Computer-Aided Design* 2023; 158: 103498. <https://doi.org/10.1016/j.cad.2023.103498>.
- [14] Zhang T, Hu G, Yang Y, et al. A Super-Resolution Reconstruction Method for Shale Based on Generative Adversarial Network. *Transport in Porous Media* 2023; 150(2): 383-426.
- [15] Zhang T, Ni M, Guan Q, et al. Reconstruction of three-dimensional porous media using multi-scale generative adversarial networks. *Journal of Applied Geophysics* 2023; 213: 105042.
- [16] Zhang T, Zhu P, Lu F. Stochastic reconstruction of porous media based on attention mechanisms and multi-stage generative adversarial network. *Computational Geosciences* 2023; 27: 515–536.
- [17] Xu K, Ba J, Kiros R, et al. Show, attend and tell: Neural image caption generation with visual attention international conference on machine learning. PMLR 2015; 2048-2057.
- [18] Itti L, Koch C. Computational modelling of visual attention. *Nature reviews neuroscience* 2001; 2(3): 194-203.
- [19] Shaham TR, Dekel T, Michaeli T. Singan: Learning a generative model from a single natural image *Proceedings of the IEEE/CVF international conference on computer vision* 2019; 4570-4580.
- [20] Hinz T, Fisher M, Wang O, et al. Improved techniques for training single-image gans *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* 2021; 1300-1309.
- [21] Misra D, Nalamada T, Arasanipalai AU, et al. Rotate to attend: Convolutional triplet

attention module Proceedings of the IEEE/CVF winter conference on applications of computer vision 2021; 3139-3148.

[22] Cao W, Feng Z, Zhang D, et al. Facial expression recognition via a CBAM embedded network. *Procedia Computer Science* 2020; 174: 463-477.

[23] Gulrajani I, Ahmed F, Arjovsky M, et al. Improved training of wasserstein gans. *Advances in neural information processing systems* 2017; arXiv: 1704. 00028.

[24] Torquato S. Statistical description of microstructures. *Annual review of materials research* 2002; 32(1): 77-111. <http://dx.doi.org/10.1146/annurev.matsci.32.110101.155324>.

[25] Strebelle S. Conditional simulation of complex geological structures using multiple-point statistics. *Mathematical geology* 2002; 34: 1-21.

[26] Mahmud K, Mariethoz G, Caers J, et al. Simulation of Earth textures by conditional image quilting. *Water Resources Research* 2014; 50(4): 3088-3107.

[27] Krishnan S, Journel AG. Spatial connectivity: from variograms to multiple-point measures. *Mathematical Geology* 2003; 35(8): 915-925.

[28] Liu X, Shapiro V. Random heterogeneous materials via texture synthesis. *Computational Materials Science* 2015; 99: 177-189.

[29] Heusel M, Ramsauer H, Unterthiner T, et al. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems* 2017; 30. arXiv: 1706. 08500.

[30] Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the inception architecture for computer vision Proceedings of the IEEE conference on computer vision and pattern recognition 2016; 2818-2826. <https://doi.org/10.1109/CVPR.2016.308>.

[31] Dong H, Blunt M J. Pore-network extraction from micro-computerized-tomography images. *Physical review E* 2009; 80(3): 036307.

[32] Avizo. Avizo User's Guide, 9th ed. 2015.

[33] Legland D, Arganda-Carreras I, Andrey P. MorphoLibJ: integrated library and plugins for mathematical morphology with ImageJ. *Bioinformatics* 2016; 32(22): 3532-3534.

[34] Song S, Mukerji T, Hou J. Geological facies modeling based on progressive growing of generative adversarial networks (GANs). *Computational Geosciences* 2021; 25: 1251-1273. <https://doi.org/10.1007/s10596-021-10059-w>.

[35] Song S, Mukerji T, Hou J. Bridging the gap between geophysics and geology with generative adversarial networks. *IEEE Transactions on Geoscience and Remote Sensing* 2021; 60: 1-11. <https://doi.org/10.1109/TGRS.2021.3066975>.

**Declaration of interests**

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

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: